Adaptively Secure Identity-Based Broadcast Encryption With a Constant-Sized Ciphertext

Jongkil Kim, Willy Susilo, *Senior Member, IEEE*, Man Ho Au, *Member, IEEE*, and Jennifer Seberry, *Senior Member, IEEE*

***Abstract*— In this paper, we present an adaptively secure identity-based broadcast encryption system featuring constant sized ciphertext in the standard model. The size of the public key and the private keys of our system are both linear in the maximum number of receivers. In addition, our system is fully collusion-resistant and has stateless receivers. Compared with the state-of-the-art, our scheme is well optimized for the broadcast encryption. The computational complexity of decryption of our scheme depends only on the number of receivers, not the maximum number of receivers of the system. Technically, we employ dual system encryption technique and our proposal offers adaptive security under the general subgroup decisional assumption. Our scheme demonstrates that the adaptive security of the schemes utilizing a composite order group can be proven under the general subgroup decisional assumption, while many existing systems working in a composite order group are secure under multiple subgroup decision assumptions. We note that this finding is of an independent interest, which may be useful in other scenarios.**

***Index Terms*— Cryptography, public key, broadcast encryption, identity-based broadcast encryption.**

I. Introduction

BROADCAST encryption (BE) [1] is a cryptographic primitive in which multiple receivers share encrypted data with a sender. In BE, a sender chooses the set of receivers, adaptively, and encrypts secret data for them. The encrypted data only can be decrypted by recipients included in the set of receivers. BE has many practical applications such as secure databases and Digital Right Management (DRM) systems including DVD and Pay TV solutions.

The security of BE is defined by the security model it follows. A BE scheme is adaptive secure [2] if it allows the adversary to declare the set that he/she wants to attack by using the public parameters and private keys compromised under the restriction that the adversary cannot possess any decryption key of the users in the target set. The selective security [3], by comparison, requires that the adversary to decide the target set before the system parameters are chosen. Selective security is a weaker notion but it is relatively easier to achieve.

Broadcast encryption was extended to identity-based broadcast encryption (IBBE) [4], [5] in which each receiver is identified by his/her unique identity as in an identity-based encryption (IBE) [6]. As identities are arbitrary bit-strings, an IBBE should support exponentially many users as potential receivers. This implies that for an IBBE to be practical, the size of parameters such as public parameters, private keys and ciphertexts must not be related to the total number of users in the system.

IBBE is often simplified to mID-KEM (multiple identity-based key encryption scheme) [7], [8] which is the cryptographic primitive combining identity-based encryption and mKEM (multiple-receiver key encapsulation Mechanism). In mID-KEM [9] and mKEM, multiple parties share a secret key for their future secure communications to be protected by symmetric cryptographic algorithms.

A trivial solution to broadcast is to encrypt the same message under each receiver's public key. However, this trivial solution possesses a ciphertext size linear with the number of receivers. Thus, the goal of broadcast encryption is to reduce the size. Although there are several realizations in broadcast encryption allowing polynomial users in the system of the ciphertext, achieving an IBBE scheme having efficient sized parameters remains a difficult problem because it has to support exponentially many users in the system using the limited entropy provided in public parameters.

An IBBE should satisfy several important properties. First, an IBBE scheme should be *fully collusion resistant* [10], [11]. This property requires that even if all the users collude, they should not be able to learn anything about the message if none of the colluding users is included in the set of receivers for the broadcast. The *stateless receivers* [12] property is also important for the efficiency of the system. If an IBBE scheme does not have stateless receivers, it must distribute private keys again whenever there is a change in the set of receivers.

In this paper, we introduce an adaptively secure IBBE scheme achieving a constant sized ciphertext in the standard

Manuscript received July 4, 2014; revised October 18, 2014; accepted December 24, 2014. Date of publication January 1, 2015; date of current version February 13, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shouhuai Xu. *(Corresponding author: Willy Susilo.)*

J. Kim, W. Susilo, and J. Seberry are with the Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: jk057@uowmail.edu.au; wsusilo@uow.edu.au; jennie@uow.edu.au).

M. H. Au is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong (e-mail: csallen@comp.polyu.edu.hk).

Digital Object Identifier 10.1109/TIFS.2014.2388156

TABLE I
COMPARISON BETWEEN PREVIOUS IBBE SCHEME WITH OURS

Scheme	Assumption	PK	SK	CT	Decrypt	Security	Order
GW [2]	*q-type*	$O(\sqrt{\ell})$	$O(1)$	$5\sqrt{\ell}G$	$2P + 2(\sqrt{\ell}+1)E$	Adaptive	Prime
LSW [19]	DBDH, DLIN	$O(1)$	$O(1)$	$(2r+8)G$	$r(2P+E)$	Selective	Prime
Attrapadung [23]	DBDH, DLIN	$O(\ell)$	$O(\ell)$	$9G$	$9P + \ell E$	Adaptive	Prime
Attrapadung [21]	*q-type*, SDs	$O(\ell)$	$O(\ell)$	$6G_N$	$6P + \ell E$	Adaptive	Composite
Ours	GSD	$O(\ell)$	$O(\ell)$	$4G_N$	$4P + kE$	Adaptive	Composite

ℓ: the maximum number of receivers, k: the number of receivers for a ciphertext, G: the size of a prime order group element, G_N: the size of a composite order group element, P: Pairing, E: Exponentiation

model. Our scheme allows exponentially many users in the system, but the maximum number of recipients in a broadcast is defined in the system setup. Our scheme is also fully collusion resistant and features stateless receivers. In order to prove the adaptive security of our scheme, we use the dual system encryption [13]–[15]. Our IBBE scheme achieves a constant sized ciphertext assuming only General Subgroup Decision (GSD) Assumption [16], which is static and simple.

II. PRELIMINARIES

Several existing broadcast encryption schemes [3], [13], [17], [18] achieve constant-sized ciphertext. While they are secure in the standard model, these schemes support only polynomially many users because they have parameters, such as public keys or private keys, which increase linearly with the number of total users in the system. In these systems, the users are normally labelled from 1 to n.

Gentry and Waters [2] suggested the first adaptively secure identity-based scheme having sub-linear sized ciphertext. First, they introduced an IBBE scheme in which a linear sized Tag is included in the ciphertext to allow exponentially many users in the system. Subsequently, they suggested a way to achieve sub-linear sized ciphertext by reusing Tag in the original scheme and increasing the size of other components in a ciphertext from constant to sublinear.

Lewko, Sahai and Waters [19] introduced a revocation scheme based on a revocation system [12], [20] which achieves broadcast encryption not by including users but by revoking users. The size of the parameters does not depend on the total number of users in the system. However, the size of the ciphertext linearly increases with the number of revoked users in their scheme. In addition, while its parameters do not depend on the total number of users in the system, adaptive security has been proved when it allows a polynomial number of users. The system can only be proven selective secure if exponentially many users are to be supported.

Similarly, an adaptively secure Key Policy Attribute Based Encryption (KP-ABE) scheme featuring constant-sized ciphertext and supporting exponentially-many attributes was introduced by Attrapadung [21]. As broadcast encryption is a special case of a KP-ABE of which the policy consists only of OR-gates, their scheme is also relevant to our discussion. We analyze this scheme when it works as a broadcast encryption scheme, and we find that our scheme is more efficient than this scheme. The size of the ciphertext and the number of pairing computations for the decryption of our scheme are two thirds of theirs. Also, the security of their scheme depends on some *q-type* assumptions while our scheme depends on some simple assumptions.

There are three IBBE systems using multilinear map [22]. Due to the properties of multi-linear map, they can be very efficient. However, although the number of the group elements of a ciphertext is constant, the size of the group elements is $O(log^2 N)$. Also, the security of these systems depends on some *q-type* assumptions, which is undesirable.

Attrapadung and Libert [23] introduced the first IBBE scheme having a constant sized ciphertext as an application of Inner Product Encryption (IPE). Since broadcast encryption can be interpreted as a special case having only OR-gates between recipients, broadcast encryption can be also achieved by IPE. Their scheme is constructed in a prime order group and has a constant sized ciphertext although the sizes of a private key and a public parameter of their scheme linearly increase with the size of maximum number of receivers in the system. To achieve this, they used the dual system encryption. Their scheme depends on standard assumptions (hardness of the Decision Linear Problem (*DLIN*) and the Decision Bilinear Diffie-Hellman Problem (*DBDH*)). However, their scheme is designed for IPE and is not well adapted for an IBBE system. Some important features are missing in their construction arising from this matter. The security of their system fails if only one receiver is included in a ciphertext because their n-wise independence argument does not hold. Also, their computataional complexity can be reduced if IPE is used to construct IBBE. They also achieved an adaptively secure broadcast encryption scheme by applying the dual system encryption to [24]. However, this scheme requires a subgroup decisional assumption, which cannot be reduced as General Subgroup Decision (GSD) Assumption.

We compare our scheme with the existing schemes, and the result is summarized in Table I. We note that we also use IPE for IBBE as in [23]. Nevertheless, we optimize the IPE scheme to support IBBE. Hence, in addition to a constant sized ciphertext, the computational complexity of our scheme only depends on the number of receivers for a broadcast. Also, we observe that there exists a possible failure in the security if only one receiver is included in a encryption. We provide a practical solution for this. Furthermore, the security of

our system depends only on GSD assumption. As a result, our adaptively secure IBBE features low cost decryption by achieving a constant sized ciphertext and low computational complexity for the decryption process. More importantly, our decryption algorithm only depends on the number of receivers of the ciphertext, instead of the maximum number of receivers, which is part of the system parameters. This offers a big advantage in comparison to the other schemes.

A. Our Technique

The traditional way to prove the security of broadcast encryption is using q-type assumptions and partitioning the key space by the set of identities of receivers and others [2], [3]. The dual system encryption [13], introduced by Waters, gives a break-through in security proof methodology by introducing the concept of semi-functional keys and ciphertext which are only used in the security proof. However, proving the invariance between a semi-functional key and a normal key is still challenging because the simulator can detect this correlation by generating a semi-functional ciphertext which can be decrypted only by a normal key to distinguish whether the key is a semi-functional key or a normal key.

Dual system encryption is used widely to provide security protocols including BE [13], [19], [25], [26].

Lewko and Waters [14] suggested a way to solve this problem. In their suggestion, when the algorithm generates a semi-functional ciphertext, the ciphertext is correlated with semi-functional keys. This means if a valid semi-functional key is used to decrypt a semi-functional ciphertext, the semi-functional key does not hinder decryption and works like a normal key, but this correlation between the semi-functional key and ciphertext is hidden to the adversary who cannot query a valid key for the challenge ciphertext.

Although the nominally semi-functionality is very helpful to prove the security, hiding the correlation is not trivial if the system has to support exponentially many users with limited entropy. Lewko and Waters [27] introduced the technique to overcome the shortage of randomness. To amplify the entropy, they localize semi-functional spaces by introducing ephemeral semi-functional space which is only used to prove the key invariance between a normal key and a semi-functional key. The random values, hiding the correlation between the key and the ciphertext, are only used in ephemeral semi-functional space. Then, the semi-functional spaces share only random values which do not interrupt to hide this correlation in ephemeral semi-functional space.

We prove the security of our scheme similarly with [27]. However, we prove the adaptive security of our system using General Subgroup Decision (GSD) Assumption [16] only. Specifically, in [27], when they proved the semi-functional invariance of their scheme, they used an assumption which cannot be reduced to GSD. In contrast, we prove semi-functional invariance without this assumption. Hence, the security of our scheme relies on fewer assumptions than Lewko and Waters' scheme [27].

Our IBBE scheme achieves adaptive security by combining dual system encryption [13] with n-wise pairwise independence argument [23], However, the n-wise independence argument does not hold if only one receiver is included in the system. Hence, first we restrict our scheme so that the number of receivers is larger than 1. Then, we provide a practical way to overcome this restriction. The computational complexity of the decryption algorithm of our scheme only depends on the number of receivers.

B. Broadcast Encryption Systems

Our broadcast encryption scheme consists of four algorithms, namely, setup (**Setup**), private key generation (**KeyGen**), encryption (**Enc**) and decryption (**Dec**) as defined below.

- **Setup**(λ, n, ℓ) takes as input the number of receivers (n) and the maximal size of a broadcast recipient group (ℓ ($\leq n$)). It outputs a public/master secret key pair $\langle PK, MSK\rangle$.
- **KeyGen**(i, MSK) takes as input an index $i \in \{1, \ldots, n\}$ and the secret key MSK. It outputs a private key d_i.
- **Enc**(S, M, PK) takes as input a subset $S \subseteq \{1, \ldots, n\}$, a message M and a public key PK. If $|S| \leq \ell$, it outputs a CT.
- **Dec**(S, i, d_i, CT, PK) takes as input a subset $S \subseteq \{1, \ldots, n\}$ an index $i \in \{1, \ldots, n\}$, a private key d_i for i, a ciphertext CT, and the public key PK. If $|S| \leq \ell$ and $i \in S$, then the algorithm outputs the message M.

Correctness For the correctness, the following property must be satisfied.

For $S \subseteq \{1, \ldots, n\}$ where $|S| \leq \ell \leq n$, let $(PK, MSK) \leftarrow Setup(\lambda, n, \ell)$, $d_i \leftarrow KeyGen(i, MSK)$ for $i \in [1, n]$ and $CT \leftarrow Enc(S, M, PK)$. Then, if $i \in S$, $Dec(S, i, d_i, CT, PK) = M$.

It should be noted that the definition of BE above is general enough to describe IBBE.

C. Security Definition

We define the adaptive security model of IBBE. This basically follows the adaptive security model of [2]. The only difference being we adapt it for an ordinary IBBE scheme while the adaptive security model of [2] is for a key encapsulation scheme.

Both the adversary and the challenger are given as input ℓ and n, i.e., the maximal size of a set of receivers S and the maximum users in a system, respectively.

- **Setup**: The challenger runs **Setup**(λ, n, ℓ) to obtain a public key PK. It gives $\mathcal{A}$ the public key PK.
- **Phase I**: The adversary $\mathcal{A}$ adaptively issues private queries for identities $i \in \{1, \ldots, n\}$.
- **Challenge**: If **Phase I** is over, The attacker declares two equal length message M_0 and M_1 and a challenge set S^* where $S^* \subseteq \{1, \ldots, n\}$ and the identities of S^* never have been queried in Phase I. If $|S^*|$ is larger than ℓ, it outputs $\perp$. Otherwise, the challenger randomly selects $b \leftarrow \{0, 1\}$ and runs encryption algorithm to obtain $CT =$ **Enc**(S^*, M_b, PK). The challenger returns CT to $\mathcal{A}$.
- **Phase II**: The adversary $\mathcal{A}$ adaptively issues private queries as **Phase I** except that added restriction that identities $i \notin S$.

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

We define the advantage of an adversary $\mathcal{A}$ in attacking the identity based broadcast encryption system IBBE with inputs (n, ℓ, λ):

$$Adv_{\mathcal{A},\mathcal{IBBE},n,\ell}(\lambda) := |Pr[b = b'] - 1/2|$$

We define that an identity based encryption system IBBE is adaptively secure if $Adv_{\mathcal{A},\mathcal{IBBE},n,\ell}(\lambda) = \epsilon$ is negligible for all PPT algorithms $\mathcal{A}$.

D. Composite Order Bilinear Groups

We briefly describe the important properties of composite order bilinear groups which were introduced in [28]. Let $\mathcal{G}$ be a group generation algorithm taking a security parameter λ as input and outputting a description of a bilinear group G. For our purposes, we will have $\mathcal{G}$ output (p_1, p_2, p_3, G, G_T, e) where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1p_2p_3$, and $e : G^2 \to G_T$ is a map such that:

1. (Bilinear) $\forall g, h \in G$, $a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$
2. (Non-degenerate) $\exists g \in G$ such that $e(g, g)$ has order N in G_T.

We assume that the group operations in G in G_T as well as the bilinear map e are computable in polynomial time with respect to λ and that the group descriptions of G and G_T include generators of the respective cyclic groups. We let G_{p_1}, G_{p_2} and G_{p_3} denote the subgroup of order p_1, p_2 and p_3 in G respectively. We note that when $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ for $i \neq j$, $e(h_i, h_j)$ is the identity element in G_T (i.e. $e(h_1, h_2) = 1$). This orthogonal property of $G_{p_1}, G_{p_2}, G_{p_3}$ will be used to implement semi-functionality in our constructions.

E. Complexity Assumption

Our scheme is adaptively secure under General Subgroup Decision (GSD) assumption [16]. To avoid duplicate statements in the security proof and demonstrate which GSD instances were used clearly, we include Assumptions 1, 2 and 3 which are special cases of GSD.

General Subgroup Decision (GSD) Assumption [16]: Let $\mathcal{G}(1^\lambda)$ be a group generator and Z_0, Z_1, ..., Z_k be a collection of non-empty subset of $\{1, 2, 3\}$ where each Z_i for $i \geq 2$ satisfies either (1) or (2) following

$$Z_0 \cup Z_i \neq \emptyset \quad \text{and} \quad Z_1 \cap Z_i \neq \emptyset \tag{1}$$

$$Z_0 \cap Z_i = \emptyset \quad \text{and} \quad Z_1 \cap Z_i = \emptyset \tag{2}$$

Then, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}(1^\lambda),$$
$$g_{Z_2} \xleftarrow{R} G_{Z_2}, \ldots, g_{Z_k} \xleftarrow{R} G_{Z_k}$$
$$D = (\mathbb{G}, g_{Z_2}, \ldots, g_{Z_k}),\ T_1 \xleftarrow{R} G_{Z_0},\ T_2 \xleftarrow{R} G_{Z_1}.$$

With the fixed collection of sets $Z_0, \ldots, Z_k$, we define the advantage of an algorithm $\mathcal{A}$ in breaking this assumption to be:

$$Adv^{GSD}_{\mathcal{G},\mathcal{A}}(\lambda) := |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|.$$

We define three assumptions as special cases of GSD assumption.

For each assumption, given a group generator $\mathcal{G}(1^\lambda)$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \mathcal{G}(1^\lambda),$$

Assumption 1 (A Special Case of GSD Assumption With $Z_0 = \{1, 2\}$ $Z_1 = \{1\}$):

$$g \xleftarrow{R} G_{p_1}, D = (\mathbb{G}, g), T_1 \xleftarrow{R} G_{p_1p_2}, T_2 \xleftarrow{R} G_{p_1}$$

Assumption 2 (A Special Case of GSD Assumption With $Z_0 = \{1\}$ $Z_1 = \{1, 3\}$):

$$g,\ X_1 \xleftarrow{R} G_{p_1},\ g_2 \xleftarrow{R} G_{p_2},\ X_3 \xleftarrow{R} G_{p_3}$$
$$D = (\mathbb{G}, g, g_2, X_1X_3),\ T_1 \xleftarrow{R} G_{p_1},\ T_2 \xleftarrow{R} G_{p_1p_3}$$

Assumption 3 (A Special Case of GSD Assumption With $Z_0 = \{1, 3\}$ $Z_1 = \{1, 2, 3\}$):

$$g,\ X_1 \xleftarrow{R} G_{p_1},\ X_2,\ Y_2 \xleftarrow{R} G_{p_2},\ g_3,\ Y_3 \xleftarrow{R} G_{p_3}$$
$$D = (\mathcal{G}, g, g_3, X_1X_2, Y_2Y_3),\ T_1 \xleftarrow{R} G_{p_1p_3},\ T_2 \xleftarrow{R} G$$

In some lemmas, the roles of p_2 and p_3 of Assumption 3 are reversed.

III. Our IBBE Construction

A. Construction

Let i be an identity of a user in the system, and S be a set of identities of recipients for a broadcast. Also we define the maximum number of receivers ℓ. We restricted the number of receivers to be greater than 1.

- Setup(λ, ℓ, n) The setup algorithm takes in n, ℓ and the security parameter λ as input. Then, it chooses a bilinear group G of order $N = p_1p_2p_3$ where p_1, p_2 and p_3 are distinct primes. Then the algorithm generates $g, u, w, v, h \xleftarrow{R} G_{p_1}$ where G_{p_i} is a subgroup of G of order p_1, and also generates randomly $MSK = \{\delta\}$ in $\mathbb{Z}_N$. It outputs
$$PK = \langle g, u, w, v^{\alpha_j}, h^{\alpha_j}, e(g, h)^\delta : j \in [0, \ell]\rangle$$
- KeyGen(MSK, PK, i) Generate $y_i, r_i \xleftarrow{R} \mathbb{Z}_N$ for identity i, randomly and the sets $\vec{X} := (x_\ell, \ldots, x_1, x_0) = (i^\ell, \ldots, i, i^0)$. Using MSK and PK, it sets
$$d_i := \langle K_0, K_1, K_2, K_{3,j} : j \in [1, \ell]\rangle$$
$$= \langle g^\delta w^{y_i}, h^{y_i}, v^{y_i} \cdot u^{r_i}, h^{(-\alpha_0 x_j/x_0 + \alpha_j)r_i} : j \in [1, \ell]\rangle$$
It should be noted that x_0 is fixed as $i^0 = 1$. However, we leave it in the definition to clarify the correctness.
- Enc(PK, M, S) First, the encryption algorithm parses S as $\{i_1, \ldots, i_k\}$ and sets $\vec{Z} = (z_\ell, .., z_0)$ where z_j is an coefficient of z^j of $\prod_{j=1}^{k}(z - i_j)$. With randomly generated $s, t \xleftarrow{R} \mathbb{Z}_N$, To output CT, it sets
$$CT := \langle C, C_0, C_1, C_2, C_3\rangle$$
$$= \langle M \cdot e(g, h)^{\delta s}, h^s, w^s v^{\langle \vec{\alpha}, \vec{Z}\rangle t}, h^{\langle \vec{\alpha}, \vec{Z}\rangle t}, u^t\rangle$$
where $\vec{\alpha} = (\alpha_\ell, \ldots, \alpha_0)$.

TABLE II
THE RECOMMENDED SIZE OF THE PARAMETERS FOR 3 PRIMES [29]

Equiv.	Min.		Max.	
	$log\, p_i$	$log\, N$	$log\, p_i$	$log\, N$
AES-128	882	2646	1075	3225
AES-196	2299	6897	2640	7920
AES-256	4614	13842	5129	15387

- Decrypt(S, i, d_i, CT, PK) Suppose $i \in S$, and calculate $\vec{Z}$, the decryption algorithm outputs

$$D = \frac{e(K_0, C_0)e(K_2, C_2)}{e(K_1, C_1)e((K_{3,1})^{z_1} \cdots (K_{3,k})^{z_k}, C_3)} = e(g, h)^{\delta s}$$

Then, it outputs a message $M = C/D$.

Correctness D can be computed as follows:

$$\begin{aligned}
E &= \frac{e(K_0, C_0)e(K_2, C_2)}{e(K_1, C_1)} = \frac{e(g^{\delta} w^{y_i}, h^s)e(v^{y_i} u^{r_i}, h^{\langle \vec{a}, \vec{Z} \rangle t})}{e(h^{y_i}, w^s v^{\langle \vec{a}, \vec{Z} \rangle t})} \\
&= \frac{e(g, h)^{s\delta} e(h, w)^{s y_i} e(h, v)^{\langle \vec{a}, \vec{Z} \rangle t y_i} e(h, u)^{t r_i \langle \vec{a}, \vec{Z} \rangle}}{e(h, w)^{s y_i} e(h, v)^{\langle \vec{a}, \vec{Z} \rangle t y_i}} \\
&= e(g, h)^{s\delta} e(h, u)^{t r_i \langle \vec{a}, \vec{Z} \rangle} \\
F &= e((K_{3,1})^{z_1} \cdots (K_{3,k})^{z_k}, C_3) \\
&= e(\prod_{j=1}^{k} (h^{(z_j(-\alpha_0 x_j / x_0 + \alpha_j)})^{r_i}, u^t) \\
&= e((h^{-\alpha_0 (\Sigma_{j=1}^{k} z_j x_j)/x_0 + \Sigma_{j=1}^{k} z_j \alpha_j})^{r_i}, u^t) \\
&= e(h^{(\alpha_0 z_0 + \Sigma_{j=1}^{k} \alpha_j z_j) r_i}, u^t) \\
&= e(h, u)^{t r_i \langle \vec{a}, \vec{Z} \rangle}
\end{aligned}$$

As i is a root of $\prod_{j=1}^{k}(z - i_j)$, $\langle \vec{X}, \vec{Z} \rangle = \Sigma_{j=0}^{k} x_j z_j = 0$, this also implies that $\Sigma_{j=1}^{k} x_j z_j = -z_0 x_0$. Therefore, $D = E/F = e(g, h)^{s\delta}$.

We restricted our scheme to have $|S|$. However, this can be accommodated by reserving one identity when system sets up and including this identity if encryption body want to share a secret with only one user. It should be noted that the private key for this reserved identity must not be given to any user.

B. Choice of Parameters

The size of parameters is determined by the security level which a broadcast system aims to achieve. In our construction, N is the product of three primes. The factors of N must not be revealed to the attackers. We recommend the size of N based on the result of Guillevic [29] in Table II for achieving equivalent security levels with AES. The sizes are calculated based on the attacks "Number Field Sieve attack" and "Elliptic Curve Method attack" [30]. The minimum of the size of parameters is calculated based on the *cost (time) equivalence*, while the maximum of the size of parameters is computed based on the *computational equivalence* [30].

IV. SECURITY ANALYSIS

In order to prove the security of our scheme, the dual system encryption was used. The security can be proved by the invariances of security games.

A. Security Properties for the Dual System Encryption IBBE

Before we present the security proof of our construction, we define semi-functional keys and a semi-functional ciphertext which are not used in the real system, but necessary in the proof. In the definition, g_2, g_3 denotes generators of G_2, G_3, respectively. In order to create semi-functional keys, we generate $\psi, \sigma \xleftarrow{R} \mathbb{Z}_N$, first. These are shared parameters in semi-functional keys regardless of the identity of i.

Semi-Functional Key: Let $(K'_0, K'_1, K'_2, K'_{3,j} : j \in [1, \ell])$ be a normal key generated by using the key generation algorithm. Then, we randomly select $\tilde{y}_i \xleftarrow{R} \mathbb{Z}_N$ for the identity i and define a semi-functional key as below

$$\begin{aligned}
K_0 &= K'_0 (g_2 g_3)^{\psi \tilde{y}_i}, \quad K_1 = K'_1 (g_2 g_3)^{\tilde{y}_i} \\
K_2 &= K'_2 (g_2 g_3)^{\sigma \tilde{y}_i}, \quad K_{3,j} = K'_{3,j} : j \in [1, \ell].
\end{aligned}$$

Semi-Functional Ciphertext: Let C', C'_0, C'_1, C'_2 and C'_3 be a properly distributed normal ciphertext. Then, with randomly generated $a, b \leftarrow \mathbb{Z}_N$, a semi-functional key is defined as below

$$C = C', \; C_0 = C'_0 g_2^a, \; C_1 = C'_1 g_2^b, \; C_2 = C'_2, \quad C_3 = C'_3$$

Semi-functional keys are only able to decrypt a normal ciphertext but not a semi-functional ciphertext although normal keys can decrypt both a normal and a semi-functional ciphertext. Now, we will prove that no PPT algorithm distinguishes the following security games with non-negligible advantage.

$Game_{Real}^{IBBE}$ This is a real game following the adaptive security model of IBBE. All private keys and the challenge ciphertext are also normal.

$Game_{k}^{IBBE}$ This is identical with $Game_{Real}^{IBBE}$ except for the types of private keys and a ciphertext. In this game, the first k keys are semi-functional keys, and the rest of the keys are normal keys and the challenge ciphertext is semi-functional.

$Game_{Final'}^{IBBE}$ This is identical with $Game_{q}^{IBBE}$ where q is the total number of key queries besides the private keys. In this game, random elements from G_{p_3} are added to K_2, $K_{3,1}, \ldots K_{3,\ell}$ components of all semi-functional keys.

$Game_{Final''}^{IBBE}$ This is identical with $Game_{Final'}^{IBBE}$ besides the challenge ciphertext. In this game, the challenge ciphertext is similar to the semi-functional ciphertext, but all components except C have additional random elements from from G_{p_3}.

$Game_{Final}^{IBBE}$ This is identical with $Game_{Final''}^{IBBE}$ besides the challenge ciphertext. In this game, the first component C of the challenge ciphertext is replaced by a random element from G_T.

Theorem 1: Our IBBE system is adaptively secure under General Subgroup Decision Assumption.

Proof: This is proved by Lemmas 1 to 7. □

Lemma 1 (Semi-Functional Ciphertext Invariance): Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{Real}^{IBBE}Adv_{\mathcal{A}} - Game_{0}^{IBBE}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.

Proof: $\mathcal{B}$ is given g_1, T. It will simulate $Game_{Real}^{IBBE}$ or $Game_{0}^{IBBE}$ with $\mathcal{A}$. It chooses random exponents $y_u, y_w, y_v, y_h, \alpha_0, \ldots, \alpha_\ell, \delta \in \mathbb{Z}_N$, and sets $g = g_1$, $u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1^{y_h}$. It publishes the public parameters:

$$PK = (g, u, w, v^{\alpha_j}, h^{\alpha_j}, e(g,h)^{\delta} : j \in [0, \ell])$$

Also, $\mathcal{B}$ generates normal keys by the key generation algorithm because it knows both PK and MSK.

In the challenge, $\mathcal{A}$ sends $\mathcal{B}$ two messages M_0, M_1 and the set of receivers, S. To make the challenge ciphertext, $\mathcal{B}$ calculates $\vec{Z} = (z_\ell, .., z_0)$ where z_j is an coefficient of z^j of $\prod_{j=1}^{k}(z - i_j)$, and implicitly sets g_1^s to be the G_{p_1} part of T (this means that T is the product of $g_1^s \in G_{p_1}$ and possibly an element of G_{p_2}). $\mathcal{B}$ also generates $t \in \mathbb{Z}_N$ randomly. It chooses $f \in \{0, 1\}$ by flipping a coin and sets:

$$C = M_f e(g^{\delta}, T^{y_h}), \quad C_0 = T^{y_h}, \quad C_1 = T^{y_w} g^{y_v \langle \vec{\alpha}, \vec{Z} \rangle t},$$
$$C_2 = g_1^{y_h \langle \vec{\alpha}, \vec{Z} \rangle t}, \quad C_3 = g_1^{y_u t}.$$

If $T \in G_{p_1}$, this is properly distributed normal ciphertext, and $\mathcal{B}$ properly simulates the $Game_{Real}^{IBBE}$. If $T \in G_{p_1 p_2}$, then we have a semi-functional ciphertext with $a = y_h s'$ and $b = y_w s'$: we denote the G_{p_2} part of T by $g_2^{s'}$ (i.e. $T = g_1^s g_2^{s'}$). Since the values of y_h, and y_w modulo p_2 are uncorrelated from their values modulo p_1, reusing the values from G_{p_1} does not correlate with the normal key. So, this is a properly distributed semi-functional ciphertext, and $\mathcal{B}$ properly simulates $Game_{0}^{IBBE}$. □

Lemma 2 (Semi-Functional Security): Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{q}^{IBBE}Adv_{\mathcal{A}} - Game_{Final}^{IBBE}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2 or Assumption 3.

Proof: This is proved by Lemmas 2.1 to 2.3. □

Lemma 2.1: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{q}^{IBBE}Adv_{\mathcal{A}} - Game_{Final'}^{IBBE}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given g_1, g_2, X_1X_3, T. It will simulate $Game_{q}^{IBBE}$ or $Game_{Final'}^{IBBE}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, \delta \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It publishes the public parameters:

$$PK = (g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, e(g,h)^{\delta} : j \in [0, \ell])$$

When $\mathcal{A}$ makes a ciphertext query by sending two messages M_0, M_1 and the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $t, s, a, b \in \mathbb{Z}_N$. Then, it randomly selects $f \in \{0, 1\}$ and returns

$$C = M_f e(g,h)^{\delta s}, C_0 = h^s g_2^a,$$
$$C_1 = w^s g_2^b v^{\langle \vec{\alpha}, \vec{Z} \rangle t}, \; C_2 = h^{\langle \vec{\alpha}, \vec{Z} \rangle t}, \; C_3 = u^t\}.$$

When $\mathcal{A}$ makes private key queries, for some identity i, $\mathcal{A}$ chooses a random $y_i', r_i' \in \mathbb{Z}_N$ and returns

$$\{(X_1X_3g_2)^{y_w y_i'}, \; (X_1X_3g_2)^{y_i'},$$
$$(X_1X_3g_2)^{y_v y_i'} T^{y_u r_i'}, \; T^{r_i'(-a_0 x_j / x_0 + a_j)} : j \in [1, \ell]\}.$$

We let $g_1^{y_{x_1}} g_2^{y_{x_2}}$ denote X_1X_3. Then, y_i equals to $y_{x_1} y_i'$ modulo p_1 and $\tilde{y}_i$ equals to y_i' modulo p_2 and $y_{x_3} y_i'$ modulo p_3. Also, r_i equals to tr_i' modulo p_1 if we write the G_{p_1} of T as g_1^t. Also it impicitly sets that $\psi = y_w$ and $\sigma = y_v$ modulo p_2, p_3. If $T \in G_{p_1}$, this has simulated $Game_{q}^{IBBE}$, properly. Also, If $T \in G_{p_1 p_3}$, because y_u and a_j modulo p_3 do not appear anywhere else, the random elements of G_{p_3} are added in K_2 and $K_{3,j}$ to each key and randomized by r_i'. Hence, this has well simulated $Game_{Final'}^{IBBE}$. □

Lemma 2.2: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{Final'}^{IBBE}Adv_{\mathcal{A}} - Game_{Final''}^{IBBE}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given g_1, g_3, X_1X_2, T. It will simulate $Game_{Final'}^{IBBE}$ or $Game_{Final''}^{IBBE}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, \delta, \psi, \sigma \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It publishes the public parameters:

$$PK = (g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, e(g,h)^{\delta})$$

When $\mathcal{A}$ makes private key queries, for some identity i, $\mathcal{B}$ chooses a random $y_i, r_i', \gamma_0', \ldots, \gamma_\ell' \in \mathbb{Z}_N$ and it returns

$$K_0 = (X_1X_2)^{y_w y_i'} g_3^{\psi y_i'}, \; K_1 = (X_1X_2g_3)^{y_i'},$$
$$K_2 = (X_1X_2)^{y_v y_i'} g_3^{\sigma y_i'} u^{r_i} (g_3)^{r_i \gamma_0'},$$
$$K_{3,j} = h^{r_i(-\alpha_0 x_j / x_0 + \alpha_j)} (g_3)^{r_i \gamma_j'} : j \in [1, \ell]\}$$

We let $g_1^{y_{x_1}} g_2^{y_{x_2}}$ denote X_1X_2. Then, y_i equals to $y_{x_1} y_i'$ modulo p_1 and $\tilde{y}_i$ equals to $y_{x_2} y_i'$ modulo p_2 and y_i' modulo p_3. So, these are properly distributed semi-functional keys.

When $\mathcal{A}$ makes a ciphertext query by sending M_0, M_1 and the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $t', t'', t''' \in \mathbb{Z}_N$ Then, it randomly selects $f \in \{0, 1\}$ and returns

$$C = M_f e(g,T)^{\delta}, C_0 = T g_2^a,$$
$$C_1 = T^{y_w} g_2^{b y_v \langle \vec{\alpha}, \vec{Z} \rangle t'''} v^{\langle \vec{\alpha}, \vec{Z} \rangle t'},$$
$$C_2 = T^{\langle \vec{\alpha}, \vec{Z} \rangle t'''} h^{\langle \vec{\alpha}, \vec{Z} \rangle t'}, \; C_3 = T^{y_u t'''} g^{y_u t'}.$$

We denote the G_{p_1} part of T as g_1^{τ}. This implicitly sets $s = \tau$ and $t = t' + \tau t'''$ modulo p_1.

If $T \in G_{p_1}$, this $\mathcal{B}$ has properly simulated $Game_{Final'}^{IBBE}$. If $T \in G_{p_1 p_3}$, we must argue that the G_3 terms attached to the ciphertext are uniformly random in order to claim that $\mathcal{B}$ simulates properly $Game_{Final''}^{IBBE}$. Let us denote by G_3 the part of ciphertext $g_3^{t_0}$, $g_3^{t_1}$, $g_3^{t_2}$ and $g_3^{t_3}$. If we also denote by G_3 the part of T as $g_3^{\tau''}$, then $t_0 = \tau''$, $t_1 = \tau''(y_w + y_v \langle \vec{\alpha}, \vec{Z} \rangle t''')$, $t_2 = \tau''(\langle \vec{\alpha}, \vec{Z} \rangle t''')$ and $t_3 = \tau''(y_u t''')$ modulo p_3. Because $\alpha_j\rangle$, y_w, y_v, y_u do not appears any G_{p_3} parts in this simulation. So, the G_3 parts of the challenge ciphertext are randomly distributed. Hence, it has simulated $Game_{Final''}^{IBBE}$. □

Lemma 2.3: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game^{IBBE}_{Final''} Adv_{\mathcal{A}} - Game^{IBBE}_{Final} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: $\mathcal{B}$ is given $g_1, g_3, X_1X_2, Y_2Y_3, T$. It will simulate $Game^{IBBE}_{Final''}$ or $Game^{IBBE}_{Final}$ using $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}$ and $h = g_1$. It publishes the public parameters:

$$PK = (g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, e(T^{y_g}, g_1))$$

When $\mathcal{A}$ makes private key queries, for some identity i, $\mathcal{B}$ chooses a random $y'_i, r'_i, \gamma'_0, \ldots \gamma'_\ell \in \mathbb{Z}_N$ and returns

$$K_0 = T^{(y_w+y_g)} w^{y'_i} (Y_2Y_3)^{(y_w+y_g)y'_i},\ K_1 = T h^{y'_i} (Y_2Y_3)^{y'_i},$$
$$K_2 = T^{y_v} v^{y'_i} (Y_2Y_3)^{y_v y'_i} u^{r_i} g_3^{r_i\gamma'_0},$$
$$K_{3,j} = h^{r_i(-\alpha_0 x_j/x_0+\alpha_j)} g_3^{r_i\gamma'_j} : j \in [1, \ell]$$

If we write Y_2Y_3 and the $G_{p_1p_3}$ part of T as $g_1^{y_{y_1}} g_3^{y_{y_3}}$ and $g_1^{\delta} g_3^{\delta''}$, respectively, this implicitly sets $y_i = \delta + y'_i$ modulo p_1. Also, $\tilde{y}'_i$ equals to $y_{y_3} y'_i + \delta''$ modulo p_3. $\psi = y_w + y_g$ modulo p_2 and p_3, and $\sigma = y_v$ modulo p_2 and p_3. If $T \in G_{p_1p_3}$, $\tilde{y}'_i$ equals to $y_{y_2} y'_i$ modulo p_2. If $T \in G$, $\tilde{y}'_i$ equals to $y_{y_3} y'_i + \delta'$ modulo p_2 if we write the G_{p_2} part of T as $g_2^{\delta'}$.

When $\mathcal{A}$ makes a ciphertext query by sending M_0, M_1 and the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $a', b', s', t \in \mathbb{Z}_N$ and returning

$$C = e(T^{y_g}, X_1X_2)^{s'}, C_0 = (X_1X_2)^{s'} (Y_2Y_3)^{a'},$$
$$C_1 = (X_1X_2)^{y_w s'} (Y_2Y_3)^{b'} v^{\langle \vec{a}, \vec{Z} \rangle t},$$
$$C_2 = h^{\langle \vec{a}, \vec{Z} \rangle t} g_3^{t'},\ C_3 = u^t g_3^{t''}$$

The random values are properly added into the $G_{p_2p_3}$ part of the ciphertext because of a', b', t', t''. If $T \in G_{p_1p_3}$, this properly simulated $Game^{IBBE}_{Final''}$. If $T \in G$, $e(g_2, g_2)^{\delta' y_{x_2} s'}$ additionally added to C of the ciphertext. It should be noted that the value of s' modulo p_2 appears C_0 and C_1 in the ciphertext, but its value is not revealed because of a' and b' modulo p_2. Hence, $e(g_2, g_2)^{\delta' y_{x_2} s'}$ is uniformly random to $\mathcal{A}$, and this has well simulated $Game^{IBBE}_{Final}$. □

B. Semi-Functional Key Invariance

It is quite challenging to prove that there is no polynomial time algorithm $\mathcal{B}$ to distinguish between $Game^{IBBE}_{k-1}$ and $Game^{IBBE}_{k}$ with non-negligible advantage because there is no restriction on $\mathcal{B}$. Hence it can generate a semi-functional ciphertext to test whether the kth key is semi-functional or normal by decrypting the semi-functional ciphertext using the kth key. In order to avoid this potential paradox, we designed oracles which output the challenge ciphertext and the private key unless the identities of the keys requested do not belong to the set of the recipients' identities of the challenge ciphertext. However, constructing these oracles and proving the invariance between them is still challenging when we work with exponentially many users because we often have to amplifying the randomness of system with the limited entropy of public keys. Hence, we defined additionally *ephemeral* key and ciphertext which are similar with the *ephemeral semi-functional* key and ciphertext introduced in [27].

In this setting, an ephemeral key decrypts both a normal and a semi-functional ciphertext, but an ephemeral challenge ciphertext is decrypted only by a normal key.

Ephemeral key: Let K'_0, K'_1, K'_2, and $K'_{3,j}$ be a normal key generated by using the key generation algorithm. With random $\gamma_0, \gamma_1, \ldots \gamma_\ell \xleftarrow{R} \mathbb{Z}_N$

$$K_0 = K'_0,\ K_1 = K'_1,\ K_2 = K'_2 (g_2g_3)^{\gamma_0},$$
$$K_{3,j} = K'_{3,j} (g_2g_3)^{\gamma_j} : j \in [1, \ell]$$

Ephemeral ciphertext: Let C', C'_0, C'_1, C'_2 and C'_3 be a properly distributed normal ciphertext. Then with random $a, b, \alpha'_0, \ldots, \alpha'_k, t', t'' \xleftarrow{R} \mathbb{Z}_N$, and outputs

$$C = C',\ C_0 = C'_0 g_2^a,$$
$$C_1 = C'_1 g_2^b g_2^{\sigma \langle \vec{a}', \vec{Z} \rangle t'},\ C_2 = C'_2 g_2^{\langle \vec{a}', \vec{Z} \rangle t'},\ C_3 = C'_3 g_2^{t''}$$

where $\vec{\alpha}' = (\alpha'_0, .. \alpha'_k)$.

It should be noted that an ephemeral ciphertext has the parameter σ shared with the semi-functional key.

1) Sequence of Games: In order to prove the invariance between $Game^{IBBE}_{k-1}$ and $Game^{IBBE}_{k}$, we additionally define security games having an ephemeral key and/or an ephemeral ciphertext and the added restriction in modulo p_3.

$Game^{IBBE'}_{k-1}$ This game is identical with $Game^{IBBE}_{k-1}$, except for the added restriction that the identity of the (k-1)th key cannot be equal to any of the identities of the challenge ciphertext modulo p_3.

$Game^{EK}_{k}$ In this game, the ciphertext is semi-functional and the kth key is ephemeral. The additional restriction on the identities modulo p_3 is retained in this game.

$Game^{EC}_{k}$ In this game, both the ciphertext the kth key are ephemeral. The additional restriction on the identities modulo p_3 is retained in this game.

$Game^{IBBE'}_{k}$ This game is identical with $Game^{IBBE}_{k}$, except for the additional restriction on the identities modulo p_3.

First, we will prove that $Game^{IBBE}_{k} \approx Game^{IBBE'}_{k}$. Then, the steps $Game^{IBBE'}_{k-1} \approx Game^{EK}_{k}$, $Game^{EK}_{k} \approx Game^{EC}_{k}$, $Game^{EC}_{k} \approx Game^{IBBE'}_{k}$ and $Game^{IBBE'}_{k} \approx Game^{IBBE}_{k}$ will be followed.

Lemma 3: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game^{IBBE}_{k} Adv_{\mathcal{A}} - Game^{IBBE'}_{k} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2 or Assumption 3.

Proof: We suppose there exists a PPT attacker $\mathcal{A}$ that distinguishes between $Game^{IBBE}_{k}$ and $Game^{IBBE'}_{k}$ with non-negligible advantage. Because $\mathcal{A}$ has non-negligible advantage, it produces two values $\mathcal{I}, \mathcal{I}' \in \mathbb{Z}_N$ which satisfy $\mathcal{I} \neq \mathcal{I}'$ modulo N but $\mathcal{I} = \mathcal{I}'$ modulo p_3 with non-negligible probability while it is simulating $Game^{IBBE}_{k}$. We set A as the g.c.d of $\mathcal{I} - \mathcal{I}'$ and N and B as N/A. Then, p_3 is divisible by A, and $B \neq 1$. There are two possible cases: 1) p_1 is divisible by B and 2) $A = p_1p_3$, $B = p_2$. The rest of the proof can be described as the same manner of [14] and [27]. The case 1 can be used to break Assumption 2, and the case 2 can be used to break Assumption 3. □

TABLE III
THE SUMMARY OF ORACLES

Oracles	Challenge key	Challenge CT	Simulating Game
$\mathbf{O}_0$	Normal	Semi-functional	$Game(IBBE, k-1)$
$\mathbf{O}_1$	Ephemeral	Semi-functional	$Game(EK, k-1)$
$\mathbf{O}_2$	Ephemeral	Ephemeral	$Game(EC, k-1)$
$\mathbf{O}_3$	Semi-functional	Semi-functional	$Game(IBBE, k)$

Game(a,b): Game_b^a

2) Oracle Lemmas: The invariance between $Game_{k-1}^{IBBE'}$ and $Game_k^{IBBE'}$ will be proved by using the oracle lemmas. In the following proofs, $\mathcal{B}$ uses oracles to simulate the security games with $\mathcal{A}$, but it cannot distinguish which oracles with which it is working. We define four oracles (O_0, O_1, O_2, O_3). Each oracle can response to an initial query, a challenge key query and a challenge ciphertext query. We summarize the relation between the oracles and the security games in Table III.

In order to respond to an initial query, the oracles randomly select $g, u, w, v, h \in G_{p_1}$ and $\alpha_0, \ldots, \alpha_\ell, s, a, \psi, \tilde{y}, y, \sigma \in \mathbb{Z}_N$ and return the group elements:

$$\{g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a, \\ w^y (g_2 g_3)^{\psi \tilde{y}}, h^y (g_2 g_3)^{\tilde{y}}, v^y (g_2 g_3)^{\sigma \tilde{y}} : j \in [0, \ell]\}.$$

The responses that each oracle outputs as a challenge key and a challenge ciphertext have different distributions according to the type of oracle. They are distributed as the following:

Oracle O_0: If the oracle receives a challenge key query for an identity $i \in \mathbb{Z}_N$, it returns the group elements which are identical with a normal key. Upon receiving a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it calculates $\vec{Z}$ for S and selects randomly $b, t \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle \vec{a}, \vec{Z} \rangle t}, h^{\langle \vec{a}, \vec{Z} \rangle t}, u^t\}.$$

Oracle O_1: If the oracle receives a challenge key query for an identity $i \in \mathbb{Z}_N$, it selects randomly $y', r', \gamma_0', \ldots, \gamma_\ell' \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^{y'}, h^{y'}, v^{y'} u^{r'} (g_2 g_3)^{\gamma_0'}, \\ h^{(-\alpha_0 x_j / x_0 + \alpha_j) r'} (g_2 g_3)^{\gamma_j'} : j \in [1, \ell]\}$$

The challenge ciphertext response is identical with O_0.

Oracle O_2: If the oracle receives a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it calculates $\vec{Z}$ for S and selects randomly $b, t, \alpha_0', \ldots, \alpha_\ell', t_1, t_2 \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle \vec{a}, \vec{Z} \rangle t} g_2^{\sigma \langle \vec{a}', \vec{Z} \rangle t_1}, h^{\langle \vec{a}, \vec{Z} \rangle t} g_2^{\langle \vec{a}', \vec{Z} \rangle t_1}, u^t g_2^{t_2}\}$$

It responses to a challenge key query in the same way as O_1.

Oracle O_3: If the oracle receives a challenge key query for an identity $i \in \mathbb{Z}_N$, it selects randomly $y', \tilde{y}', r' \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^{y'} (g_2 g_3)^{\psi \tilde{y}'}, h^{y'} (g_2 g_3)^{\tilde{y}'}, v^{y'} (g_2 g_3)^{\sigma \tilde{y}'} u^{r'}, \\ h^{(-\alpha_0 x_j / x_0 + \alpha_j) r'} : j \in [1, \ell]\}$$

The challenge ciphertext response is identical with O_0. The invariances of (O_0, O_1, O_2, O_3) are proved by several lemmas with additionally defined sub-oracles. For the overview of proving sequences, we add Table IV.

TABLE IV
THE SUMMARY OF HOPS

Lemma	Invariance between	Assumption		T in
4.1	$\mathbf{O}_0 \approx \mathrm{O}_{0.1}$	A.2	$(G_{p_1}, G_{p_1 p_3})$	Key
4.2	$\mathrm{O}_{0.1} \approx \mathbf{O}_1$	A.3	$(G_{p_1 p_3}, G_{p_1 p_2 p_3})$	Key
5.1	$\mathbf{O}_1 \approx \mathrm{O}_{1.1}$	A.2	$(G_{p_1}, G_{p_1 p_3})$	CT
5.2	$\mathrm{O}_{1.1} \approx \mathrm{O}_{1.2}$	A.3	$(G_{p_1 p_3}, G_{p_1 p_2 p_3})$	CT
5.3	$\mathrm{O}_{1.2} \approx \mathbf{O}_2$	A.2	$(G_{p_1}, G_{p_1 p_3})$	CT
6.1	$\mathbf{O}_2 \approx \mathrm{O}_{2.1}$	A.2	$(G_{p_1}, G_{p_1 p_3})$	Key
6.2	$\mathrm{O}_{2.1} \approx \mathrm{O}_{2.2}$	A.3(Rev)	$(G_{p_1 p_2}, G_{p_1 p_2 p_3})$	CT
6.3	$\mathrm{O}_{2.2} \approx \mathrm{O}_{2.3}$	A.3	$(G_{p_1 p_3}, G_{p_1 p_2 p_3})$	Key
6.4	$\mathrm{O}_{2.3} \approx \mathrm{O}_{2.4}$	A.3(Rev)	$(G_{p_1 p_2 p_3}, G_{p_1 p_2})$	CT
6.5	$\mathrm{O}_{2.4} \approx \mathrm{O}_{2.5}$	A.2	$(G_{p_1}, G_{p_1 p_3})$	CT
6.6	$\mathrm{O}_{2.5} \approx \mathrm{O}_{2.6}$	A.3	$(G_{p_1 p_2 p_3}, G_{p_1 p_3})$	CT
6.7	$\mathrm{O}_{2.6} \approx \mathrm{O}_{2.7}$	A.2	$(G_{p_1 p_3}, G_{p_1})$	CT
6.8	$\mathrm{O}_{2.7} \approx \mathrm{O}_{2.8}$	A.3	$(G_{p_1 p_2 p_3}, G_{p_1 p_3})$	Key
6.9	$\mathrm{O}_{2.8} \approx \mathbf{O}_3$	A.2	$(G_{p_1 p_3}, G_{p_1})$	Key

Lemma 4: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_0 Adv_{\mathcal{A}} - O_1 Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2 or Assumption 3.

Proof: This is proved by Lemma 4.1 and Lemma 4.2 with an additional oracle $O_{0.1}$.

Oracle $O_{0.1}$: If the oracle receives a challenge key query for an identity $i \in \mathbb{Z}_N$, it selects randomly $y', r', \gamma_0', \ldots \gamma_\ell' \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^{y'}, h^{y'}, v^{y'} u^{r'} g_3^{\gamma_0'}, h^{(-\alpha_0 x_j / x_0 + \alpha_j) r'} g_3^{\gamma_j'} : j \in [1, \ell]\}.$$

It responses to an initial query and a challenge ciphertext query in the same way as O_0. □

Lemma 4.1: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_0 Adv_{\mathcal{A}} - O_{0.1} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given $g_1, g_2, X_1 X_3, T$. It will simulate O_0 or $O_{0.1}$ using $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, \tilde{y} \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It sends the group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a, \\ (X_1 X_3)^{y_w} g_2^{y_w \tilde{y}}, (X_1 X_3) g_2^{\tilde{y}}, (X_1 X_3)^{y_v} g_2^{y_v \tilde{y}} : j \in [0, \ell])$$

If we write $X_1 X_3$ as $g_1^{y_{x_1}} g_3^{y_{x_3}}$, this implicitly sets y equal to y_{x_1} modulo p_1 and $\tilde{y}$ equal to y_{x_3} modulo p_3. Also, ψ equals y_w and σ equals y_v modulo p_2 and p_3. Because the values of y_w and y_v modulo p_1 do not correlate with their values in modulo p_2 and p_3, this is properly distributed.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S^*, $\mathcal{B}$ chooses a random $b, t \in \mathbb{Z}_N$ and returns the group elements

$$\{w^s g_2^b v^{\langle \vec{a}, \vec{Z} \rangle t}, h^{\langle \vec{a}, \vec{Z} \rangle t}, u^t\}.$$

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y' \in \mathbb{Z}_N$ and returns

$$\{w^{y'}, h^{y'}, v^{y'} T^{y_u}, T^{-\alpha_0 x_j / x_0 + \alpha_j} : j \in [1, \ell]\}.$$

This implicitly sets $g_1^{r'}$ to be the G_{p_1} part of T. If $T \in G_{p_1}$, then this matches the distribution of O_0. If $T \in G_{p_1 p_3}$,

then this matches the distribution of $O_{0.1}$. Also this implicitly sets $\alpha'_j = \alpha_j$, $\gamma'_0 = r'' y_u$ and $\gamma'_j = r''(-\alpha_0 x_j/x_0 + \alpha_j)$ modulo p_3 when we write the G_{p_3} part of T as $g_3^{r''}$. γ'_0 contains y_u modulo p_3 which does not appear anywhere else. Also, for all $j \in [1, \ell]$, γ'_j contains α_j modulo p_3 which also does not appear anywhere else. Because y_u modulo p_3 and α_j modulo p_3 are not correlated with their values in modulo p_1, this challenge ciphertext is randomly distributed. □

Lemma 4.2: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{0.1}Adv_{\mathcal{A}} - O_1 Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: $\mathcal{B}$ is given $g_1, g_3, X_1X_2, Y_2Y_3, T$. It will simulate $O_{0.1}$ or O_1 using $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, \psi, y, \tilde{y}, \sigma \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1^{y_u}$, $w = g_1^{y_w}$, $v = g_1^{y_v}$, and $h = g_1$. It sends the group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, X_1X_2, \\ w^y (Y_2Y_3)^{\psi\tilde{y}}, h^y (Y_2Y_3)^{\tilde{y}}, v^y (Y_2Y_3)^{\sigma\tilde{y}} : j \in [1, \ell])$$

This is implicitly sets $a = y_{x_2}$ modulo p_2 when we write $X_1X_2 = g_1^s g_2^{y_{x_2}}$.

When $\mathcal{A}$ makes a ciphertext-type query for the set of recivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing a random $t \in \mathbb{Z}_N$ and returning

$$\{(X_1X_2)^{y_w}, v^{\langle\vec{\alpha},\vec{Z}\rangle t}, h^{\langle\vec{\alpha},\vec{Z}\rangle t}, u^t\}.$$

This implies $b = y_w y_{x_2}$ modulo p_2. a and b are uniformly distributed because y_w modulo p_2 does not appear anywhere else.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y' \in \mathbb{Z}_N$ and returns

$$\{w^{y'}, h^{y'}, v^{y'} T^{y_u}, T^{-\alpha_0 x_j/x_0 + \alpha_j} : j \in [1, \ell]\}.$$

The G_1 part of the challenge ciphertext is properly distributed if we write $g_1^{r'}$ as the G_1 part of T. If we write the G_{p_3} part of T as $g_3^{r''}$, this implicitly sets $\gamma'_0 = r'' y_u$ modulo p_3 and $\gamma'_j = r''(-\alpha_0 x_j/x_0 + \alpha_j)$ modulo p_3. Because y_u and α_j modulo p_3 do not appear anywhere else, the G_{p_3} parts of this challenge ciphertext is randomly distributed. Hence, if $T \in G_{p_1p_3}$, then this matches the distribution of $O_{0.1}$. If $T \in G$, then this matches the distribution of O_1 because y_u and α_j modulo p_2 do not appear anywhere else and does not correlate their values in modulo p_1 and p_3, this is the properly distributed challenge ciphertext. □

Lemma 5: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_1 Adv_{\mathcal{A}} - O_2 Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2 or Assumption 3.

Proof: This is proved by Lemma 5.1, Lemma 5.2 and Lemma 5.3 with additional oracles $O_{1.1}$ and $O_{1.2}$.

Oracle $O_{1.1}$: If the oracle receives a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it selects randomly $b, t, \alpha'_0, \ldots, \alpha'_\ell, t_1, t_2 \overset{R}{\leftarrow} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle\vec{\alpha},\vec{Z}\rangle t} g_3^{\sigma\langle\vec{\alpha}',\vec{Z}\rangle t_1}, h^{\langle\vec{\alpha},\vec{Z}\rangle t} g_3^{\langle\vec{\alpha}',\vec{Z}\rangle t_1}, u^t g_3^{t_2}\}.$$

It responses an initial query and a challenge ciphertext query in the same way as O_1.

Oracle $O_{1.2}$: If the oracle receives a challenge ciphertext query for an identity $i \in \mathbb{Z}_N$, it selects randomly $b, t, \alpha'_0, \ldots, \alpha'_\ell, t_1, t_2 \overset{R}{\leftarrow} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle\vec{\alpha},\vec{Z}\rangle t} (g_2g_3)^{\sigma\langle\vec{\alpha}',\vec{Z}\rangle t_1}, \\ h^{\langle\vec{\alpha},\vec{Z}\rangle t} (g_2g_3)^{\langle\vec{\alpha}',\vec{Z}\rangle t_1}, u^t (g_2g_3)^{t_2}\}$$

It responses an initial query and a challenge ciphertext query in the same way as $O_{1.1}$. □

Lemma 5.1: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_1 Adv_{\mathcal{A}} - O_{1.1} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given g_1, g_2, X_1X_3, T. It will simulate O_1 or $O_{1.1}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_w, y_v, y_h, \sigma', \alpha_0, \ldots, \alpha_\ell, s, a, \tilde{y} \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It sends the group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a, (X_1X_3)^{y_w} g_2^{y_w\tilde{y}}, \\ (X_1X_3) g_2^{\tilde{y}}, (X_1X_3)^{y_v} g_2^{\sigma'\tilde{y}} : j \in [0, \ell])$$

We let X_1X_3 denote as $g_1^{y_{x_1}} g_3^{y_{x_3}}$. Then, this implicitly sets y equals to y_{x_1} modulo p_1. Also, ψ equal to y_w sets σ equal to σ' modulo p_2 and y_v modulo p_3.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y', r', \gamma'_0, \ldots, \gamma'_\ell \in \mathbb{Z}_N$ and returns

$$\{w^{y'}, h^{y'}, v^{y'} (X_1X_3)^{r'y_u} g_2^{\gamma'_0}, \\ (X_1X_3)^{r'(-\alpha_0 x_j/x_0 + \alpha_j)} g_2^{\gamma'_j} : j \in [1, \ell]\}$$

This implies that $\gamma'_0 = r' y_u y_{x_3}$ modulo p_3 and $\gamma'_j = r'(-\alpha_0 x_j/x_0 + \alpha_j) y_{x_3}$ modulo p_3.

When $\mathcal{A}$ makes a ciphertext-type query for the set of recivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by returning

$$\{w^s g_2^b T^{y_v\langle\vec{\alpha},\vec{Z}\rangle}, T^{\langle\vec{\alpha},\vec{Z}\rangle}, T^{y_u}\}.$$

This implicitly sets g^t to be the G_{p_1} part of T. If $T \in G_{p_1}$, then this matches the distribution of O_1 because y_u, α_j of G_{p_3} part of the challenge key does not appear anywhere else. However, if $T \in G_{p_1p_3}$, α_j modulo p_3 for $j \in [0, \ell]$ also appears in the challenge ciphertext. We must argue $-\alpha_0 x_j/x_0 + \alpha_j$ modulo p_3 for $j \in [1, \ell]$ are uniformly random even if $\langle\vec{\alpha}, \vec{Z}\rangle$ modulo p_3 for $j \in [0, \ell]$ is given: Let $\alpha'_j = \alpha_j$ modulo p_3 for all $j \in [0, \ell]$. Then, we rewrite the relations γ'_j, α'_j and $\langle\vec{\alpha}', \vec{Z}\rangle$ as follows.

$$\begin{pmatrix} -x_1/x_0 & 1 & & & \\ -x_2/x_0 & & 1 & & \\ \vdots & & & \ddots & \\ -x_k/x_0 & & & & 1 \\ z_0 & z_1 & z_2 & \cdots & z_k \end{pmatrix} \begin{pmatrix} \alpha'_0 \\ \alpha'_1 \\ \vdots \\ \alpha'_{k-1} \\ \alpha'_k \end{pmatrix} = \begin{pmatrix} \gamma'_1 \\ \gamma'_2 \\ \vdots \\ \gamma'_k \\ \langle\vec{\alpha}', \vec{Z}\rangle \end{pmatrix}$$

Because α'_j modulo p_3 is uniformly random and does not correlate their values with those in modulo p_1 by CRT, γ'_j for all $j \in [1, \ell]$ and $\langle\vec{\alpha}', \vec{Z}\rangle$ are k-wise independent for $k > 1$.

This implies that $\gamma'_1, \ldots, \gamma'_k$ are $\langle \vec{\alpha}', \vec{Z} \rangle$ are uniformly distributed. It should be noted that if $k = 1$, γ'_1 is equal to $\langle \vec{\alpha}', \vec{Z} \rangle$ because z_0 equal to $-x_1/x_0$ and $z_1 = 1$. Also, we stress that $\gamma'_{k+1}, \ldots \gamma'_\ell$ given to the adversary shares the α'_0, but the value of α'_0 is not revealed because for all $j \in [k+1, \ell]$, γ'_j has α_j which does not appear anywhere else. □

Lemma 5.2: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{1.1}Adv_{\mathcal{A}} - O_{1.2}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: $\mathcal{B}$ is given $g_1, g_3, X_1X_2, Y_2Y_3, T$. It will simulate $O_{1.1}$ or $O_{1.2}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, \psi, y, \tilde{y}, \sigma \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}$, and $h = g_1$. It sends the group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, (X_1X_2), \\ w^y(Y_2Y_3)^{\psi\tilde{y}}, h^y(Y_2Y_3)^{\tilde{y}}, v^y(Y_2Y_3)^{y_v\tilde{y}} : j \in [0, \ell])$$

This implicitly sets $a = y_{x_2}$ modulo p_2 if we write $X_1 = g_1^s g_2^{y_{x_2}}$.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y', r', \gamma'_0, \ldots, \gamma'_\ell \in \mathbb{Z}_N$ and returns

$$\{w^{y'}, h^{y'}, v^{y'}u^{r'}(Y_2Y_3)^{\gamma'_0}, \\ h^{r'(-\alpha_0 x_j/x_0+\alpha_j)}(Y_2Y_3)^{\gamma'_j} : j \in [1, \ell]\}$$

When $\mathcal{A}$ makes a ciphertext-type query for the set of recivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by returning

$$\{(X_1X_2)^{y_w}T^{y_v\langle \vec{\alpha}, \vec{Z} \rangle}, T^{\langle \vec{\alpha}, \vec{Z} \rangle}, T^{y_u}\}.$$

This implies $b = y_w y_{x_2}$. a and b modulo p_2 are uniformly distributed because y_w modulo p_2 do not appear anywhere else.

If $T \in G$, then this matches the distribution of $O_{1.2}$. If we write $(g_2g_3)^{t'}$ to be the $G_{p_2p_3}$ part of T., then this implies that $t_1 = t'$, $t_2 = t'y_u$ and $\alpha'_j = \alpha_j$ modulo p_2 and p_3 for $j \in [0, \ell]$. Because α_j, y_u modulo p_2 and p_3 do not appear anywhere else, these are properly distributed. Similarly, If $T \in G_{p_1p_3}$, then this matches the distribution of $O_{1.1}$. □

Lemma 5.3: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{1.2}Adv_{\mathcal{A}} - O_2Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given g_1, g_2, X_1X_3, T. It will simulate $O_{1.2}$ or O_2 with $\mathcal{A}$. It chooses random exponents $y_g, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, y_2 \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. Then, the responses of the initial and challenge-key queries can be generated the same way as Lemma 5.1.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ randomly selects $s, b, \alpha_0, \ldots, \alpha_\ell, t_1, t_2$ and responds to $\mathcal{A}$ by returning

$$\{w^s g_2^b T^{y_v\langle \vec{\alpha}, \vec{Z} \rangle} g_2^{\sigma'\langle \vec{\alpha}', \vec{Z} \rangle t_1}, T^{\langle \vec{\alpha}, \vec{Z} \rangle} g_2^{\langle \vec{\alpha}', \vec{Z} \rangle t_1}, Tg_2^{t_2}\}.$$

This is possible because g_2 was given. If we denote g^t to be the G_{p_1} part of T, the G_{p_1} part of the challenge ciphertext is properly distributed. If $T \in G_{p_1}$, then this matches the distribution of O_2. If $T \in G_{p_1p_3}$, this matches the distribution of $O_{1.2}$ for the same reasons as for Lemma 5.1. □

Lemma 6: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_2Adv_{\mathcal{A}} - O_3Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2 or Assumption 3.

Proof: This is proved by Lemmas 6.1 to 6.9 with additional oracles $Oracle\, O_{2.1}$, $Oracle\, O_{2.2}$, $Oracle\, O_{2.3}$, $Oracle\, O_{2.4}$, $Oracle\, O_{2.5}$, $Oracle\, O_{2.6}$, $Oracle\, O_{2.7}$ and $Oracle\, O_{2.8}$.

Oracle $O_{2.1}$: If the oracle receives a challenge key query for an identity $i \in \mathbb{Z}_N$, it selects randomly $y', \tilde{y}', r', \gamma'_0, \ldots, \gamma'_\ell \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^{y'}g_3^{\psi\tilde{y}'}, h^{y'}g_3^{\tilde{y}'}, v^{y'}u^{r'}(g_2g_3)^{\gamma'_0}g_3^{\sigma\tilde{y}'}, \\ h^{r'(-\alpha_0 x_j/x_0+\alpha_j)}(g_2g_3)^{\gamma'_j} : j \in [1, \ell]\}$$

It responses to an initial query and a challenge ciphertext query in the same way as O_2.

Oracle $O_{2.2}$: The response for an initial query is identical with that of $O_{2.1}$ except that $h^s(g_2g_3)^a$ replaces $h^s g_2^a$.

If the oracle receives a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it selects randomly $b, \alpha'_0, \ldots, \alpha'_\ell, t, t_1, t_2, t_3, t_4, t_5 \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{\sigma\langle \vec{\alpha}', \vec{Z} \rangle t_1} g_3^{t_3}, h^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{\langle \vec{\alpha}', \vec{Z} \rangle t_1} g_3^{t_4}, u^t g_2^{t_2} g_3^{t_5}\}$$

It responses to a challenge ciphertext query in the same way as $O_{2.1}$.

Oracle $O_{2.3}$: If the oracle receives a challenge key query for identity $i \in \mathbb{Z}_N$, it selects randomly $y', \tilde{y}', r', \gamma'_0, \ldots, \gamma'_\ell \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^{y'}(g_2g_3)^{\psi\tilde{y}'}, h^{y'}(g_2g_3)^{\tilde{y}'}, v^{y'}u^{r'}(g_2g_3)^{\gamma'_0+\sigma\tilde{y}'}, \\ h^{r'(-\alpha_0 x_j/x_0+\alpha_j)}(g_2g_3)^{\gamma'_j} : j \in [1, \ell]\}$$

It responds to an initial query and a challenge ciphertext query in the same way as $O_{2.2}$.

Oracle $O_{2.4}$: The response for an initial query is identical with that of $O_{2.1}$.

If the oracle receives a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it selects randomly $s, b, \alpha'_0, \ldots, \alpha'_\ell, t, t_1, t_2 \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{\sigma\langle \vec{\alpha}', \vec{Z} \rangle t_1}, h^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{\langle \vec{\alpha}', \vec{Z} \rangle t_1}, u^t g_2^{t_2}\}$$

It responds to a challenge ciphertext query in the same way as $O_{2.3}$.

Oracle $O_{2.5}$: If the oracle receives a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it selects randomly $s, b, \alpha'_0, \ldots, \alpha'_\ell, t, t_1, t_2, t_3, t_4 \xleftarrow{R} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{\sigma\langle \vec{\alpha}', \vec{Z} \rangle t_1} g_3^{\sigma t_3}, h^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{\langle \vec{\alpha}', \vec{Z} \rangle t_1} g_3^{t_3}, u^t g_2^{t_2} g_3^{t_4}\}$$

It responds to an initial query and a challenge ciphertext query in the same way as $O_{2.4}$.

Oracle $O_{2.6}$: If the oracle receives a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it selects randomly $s, b, t, t_3, t_4 \overset{R}{\leftarrow} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_3^{\sigma t_3},\ h^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_3^{t_3},\ u^t g_3^{t_4}\}$$

It responds to an initial query and a challenge ciphertext query in the same way as $O_{2.5}$.

Oracle $O_{2.7}$: If the oracle receives a challenge ciphertext query for a set of recipients $S \subset \{1, \ldots, n\}$, it selects randomly $s, b, t \overset{R}{\leftarrow} \mathbb{Z}_N$, then returns the group elements

$$\{w^s g_2^b v^{\langle \vec{\alpha}, \vec{Z} \rangle t},\ h^{\langle \vec{\alpha}, \vec{Z} \rangle t},\ u^t\}$$

It responds to an initial query and a challenge ciphertext query in the same way as $O_{2.6}$.

Oracle $O_{2.8}$: If the oracle receives a challenge key query for an identity $i \in \mathbb{Z}_N$, it selects randomly $y', \tilde{y}', r', \gamma'', \gamma''' \overset{R}{\leftarrow} \mathbb{Z}_N$, then returns the group elements

$$\{w^{y'}(g_2 g_3)^{\psi \tilde{y}'}, h^{y'}(g_2 g_3)^{\tilde{y}'}, v^{y'} u^{r'} (g_2 g_3)^{\sigma \tilde{y}'} g_3^{\gamma_0'},\ h^{r'(-\alpha_0 x_j / x_0 + \alpha_j)} g_3^{\gamma_j'} : j \in [1, \ell]\}$$

It responds to an initial query and a challenge ciphertext query in the same way as $O_{2.7}$. □

Lemma 6.1: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_2 Adv_{\mathcal{A}} - O_{2.1} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given $g_1, g_2, X_1 X_3, T$. It will simulate O_2 or $O_{2.1}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, \tilde{y} \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1^{y_u}$, $w = g_1^{y_w}$, $v = g_1^{y_v}$, and $h = g_1$. It sends the group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a,\ (X_1 X_3)^{y_w} g_2^{y_w \tilde{y}}, (X_1 X_3) g_2^{\tilde{y}}, (X_1 X_3)^{y_v} g_2^{y_v \tilde{y}} : j \in [1, \ell])$$

This implicitly sets $g_1^y = X_1$ modulo p_1. Also, ψ equals y_w and σ equals y_v modulo p_2, p_3.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $b, t, \alpha_0', \ldots, \alpha_\ell', t_1, t_2 \in \mathbb{Z}_N$ and returning

$$\{w^s g_2^b v^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{y_v \langle \vec{\alpha}', \vec{Z} \rangle t_1},\ h^{\langle \vec{\alpha}, \vec{Z} \rangle t} g_2^{\langle \vec{\alpha}', \vec{Z} \rangle t_1},\ u^t g_2^{t_2}\}$$

where $\vec{\alpha}' = (\alpha_0', \ldots, \alpha_\ell')$.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $\gamma \in \mathbb{Z}_N$ and returns

$$\{T^{y_w},\ T,\ T^{y_v}(X_1 X_3 g_2^{\gamma})^{y_u},\ (X_1 X_3 g_2^{\gamma})^{-\alpha_0 x_j / x_0 + \alpha_j} : j \in [1, \ell - 1]\}$$

If we denote $X_1 X_3 = g_1^{y_{x_1}} g_3^{y_{x_3}}$, this implicitly sets $r' = y_{x_1}$ modulo p_1. We note $\gamma_0' = \gamma y_u$ modulo p_2 and $\gamma_0' = y_{x_3} y_u$ modulo p_3, also $\gamma_j' = \gamma(-\alpha_0 x_j / x_0 + \alpha_j)$ modulo p_2 and $\gamma_j' = y_{x_3}(-\alpha_0 x_j / x_0 + \alpha_j)$ modulo p_3.

Let $T \in G_{p_1}$ and $g_1^{y'}$ be the G_{p_1} part of T, then this matches the distribution of O_2. If $T \in G_{p_1 p_3}$ $(g_1 g_3)^{y'}$ is the $G_{p_1 p_3}$ part of T, then this matches the distribution of $O_{2.1}$ because y_u and $\alpha_0, \ldots \alpha_\ell$ modulo p_2 do not appear anywhere else. □

Lemma 6.2: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.1} Adv_{\mathcal{A}} - O_{2.2} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: In this lemma G_{p_2} and G_{p_3} parts of Assumption 3 are reversed. $\mathcal{B}$ is given $g_1, g_2, X_1 X_3, Y_2 Y_3, T$. It will simulate $O_{2.1}$ or $O_{2.2}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, y, \sigma \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It sends the group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, T, w^y (Y_2 Y_3)^{y_w}, h^y (Y_2 Y_3), v^y (Y_2 Y_3)^{\sigma} : j \in [1, \ell])$$

This is properly distributed if we denote the G_{p_1} part of T by g_1^s. Also, this sets $\psi = y_w$ modulo p_2, p_3. If $T \in G_{p_1 p_2}$, this is a properly distributed set of group elements of $O_{2.1}$. If $T \in G$, this is properly distributed set of group elements of $O_{2.2}$.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $r', \gamma_0', \ldots, \gamma_\ell' \in \mathbb{Z}_N$ and it returns

$$\{(X_1 X_3)^{y_w},\ (X_1 X_3), (X_1 X_3)^{y_v} u^{r'} (Y_2 Y_3)^{\gamma_0'},\ h^{r'(-\alpha_0 x_j / x_0 + \alpha_j)} (Y_2 Y_3)^{\gamma_j'} : j \in [1, \ell]\}$$

This is properly a distributed challenge-key. It should be noted that y_v modulo p_3 was used but not revealed because there is random parameter γ_0' modulo p_3 which does not appear in any other component.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $t', t'', t''' \in \mathbb{Z}_N$ and returns

$$\{T^{y_w} T^{y_v \langle \vec{\alpha}, \vec{Z} \rangle t'''} v^{\langle \vec{\alpha}, \vec{Z} \rangle t'} g_2^{\sigma \langle \vec{\alpha}, \vec{Z} \rangle t'},\ T^{\langle \vec{\alpha}, \vec{Z} \rangle t'''} h^{\langle \vec{\alpha}, \vec{Z} \rangle t'} g_2^{\langle \vec{\alpha}, \vec{Z} \rangle t'},\ T^{y_u t'''} g^{y_u t'} g_2^{t''}\}.$$

We denote the $G_{p_1 p_2}$ part of T as $g_1^{\tau} g_2^{\tau'}$. This implicitly sets $s = \tau$ and $t = t' + \tau t'''$ modulo p_1. Also, G_2 parts of the challenge ciphertext distribute $g_2^b g_2^{\sigma \langle \vec{\alpha}, \vec{Z} \rangle t_1}$, $g_2^{\langle \vec{\alpha}, \vec{Z} \rangle t_1}$, $g_2^{t_2}$ where $b = \tau'(y_w + y_v \langle \vec{\alpha}, \vec{Z} \rangle t''' - \sigma \langle \vec{\alpha}, \vec{Z} \rangle t''')$, $t_1 = \tau' t''' + t'$, $t_2 = y_u \tau' t''' + t''$. b is not correlated with t_1 and t_2 because y_v modulo p_2 appears only here. Also, due to t' and t'', t_1 and t_2 do not correlate. Therefore, the G_2 terms here are properly distributed. If $T \in G_{p_1 p_2}$, this $\mathcal{B}$ has properly simulated $O_{2.1}$. If $T \in G$, we must argue that the G_3 terms attached to the ciphertext are uniformly random in order to claim that $\mathcal{B}$ simulates properly $O_{2.2}$. Let us denote by G_3 the part of ciphertext $g_3^{t_3}$, $g_3^{t_4}$ and $g_3^{t_5}$. If we also denote by G_3 the part of T as $g_3^{\tau''}$, then $t_3 = \tau''(y_w + y_v \langle \vec{\alpha}, \vec{Z} \rangle t''')$, $t_4 = \tau''(\langle \vec{\alpha}, \vec{Z} \rangle t''')$ and $t_5 = \tau''(y_u t''')$ modulo p_3. Neither t_3 nor t_4 correlates with t_5 because of $\langle \vec{\alpha}, \vec{Z} \rangle$ which is randomly distributed as $\langle \vec{\alpha}, \vec{Z} \rangle$ modulo p_3 do not appear anywhere. Also t_3 and t_4 do not correlate to each other because y_v does not reveal its value although it appears within the challenge key. So, the G_3 parts of the challenge ciphertext are properly distributed. □

Lemma 6.3: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.2} Adv_{\mathcal{A}} - O_{2.3} Adv_{\mathcal{A}} = \epsilon$. Then we can

construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: $\mathcal{B}$ is given $g_1, g_3, X_1X_2, Y_2Y_3, T$. It will simulate $O_{2.2}$ or $O_{2.3}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, y \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1^{y_u}$, $w = g_1^{y_w}$, $v = g_1^{y_v}$, and $h = g_1$. It sends the group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s(Y_2Y_3)^a, w^y(Y_2Y_3)^{y_w}, h^y(Y_2Y_3), v^y(Y_2Y_3)^{y_v} : j \in [0, \ell])$$

This implicitly sets $\psi = y_w$ and $\sigma = y_v$ modulo p_2 and p_3.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $b', t', t_4, t_5 \in \mathbb{Z}_N$ and returns

$$\{w^s(Y_2Y_3)^{b'}(X_1X_2)^{y_v\langle\vec{\alpha},\vec{Z}\rangle t'}, (X_1X_2)^{\langle\vec{\alpha},\vec{Z}\rangle t'} g_3^{t_4}, (X_1X_2)^{y_u t'} g_3^{y_u t_5}\}.$$

Then the G_1 part of challenge ciphertext properly is distributed and $t = t' y_{x_1}$. We write $X_1 = g_1^{y_{x_1}}$. Also, the G_2 part of challenge ciphertext, $t_1 = t'$ modulo p_2 and $t_2 = y_u t'$ modulo p_2, are properly distributed. Moreover, if we denote Y_2Y_3 a $g_2^{y_{y_2}} g_3^{y_{y_3}}$, b modulo p_2 equal to $b' y_{y_2}$. The G_3 part also properly distributed with random values, $t_3 = b' y_{y_3}$ modulo p_3, t_4 and t_5.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $r', \gamma_0', \ldots, \gamma_\ell' \in \mathbb{Z}_N$ and returns

$$\{T^{y_w}, T, T^{y_v} u^{r'} (Y_2Y_3)^{\gamma_0'}, h^{r'(-a_0 x_j/x_0 + a_j)} (Y_2Y_3)^{\gamma_j'} : j \in [1, \ell]\}$$

If $T \in G_{p_1p_3}$, the challenge key type response is identically distributed to a response from $O_{2.2}$. If $T \in G$, then the challenge key-type reponse is identically distributed to a response from $O_{2.3}$. □

Lemma 6.4: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.3}Adv_{\mathcal{A}} - O_{2.4}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: In this lemma, G_{p_2} and G_{p_3} parts of Assumption 3 are reversed. $\mathcal{B}$ is given $g_1, g_2, X_1X_3, Y_2Y_3, T$. It will simulate $O_{2.3}$ or $O_{2.4}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, \psi, y', \sigma \in \mathbb{Z}_N$, and set $g = g_1^{y_g}, u = g_1^{y_u}$, $w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. Then, initial response, normal keys can be responded by generating them as the same way of Lemma 6.2.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $r', \gamma_0', \ldots, \gamma_\ell' \in \mathbb{Z}_N$ and returns

$$\{(X_1X_3g_2)^{y_w}, (X_1X_3g_2), (X_1X_3g_2)^{y_v} u^{r'} (Y_2Y_3)^{\gamma_0'}, h^{r'(-a_0 x_j/x_0 + a_j)} (Y_2Y_3)^{\gamma_j'} : j \in [1, \ell]\}$$

This is properly distributed challenge-key. It should be noted that y_v modulo p_2 and p_3 was used but not revealed because there is random parameter r' modulo p_2 and p_3 which does not appear anywhere else.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $t', t'', t''' \in \mathbb{Z}_N$ and returning

$$\{T^{y_w} T^{y_v\langle\vec{\alpha},\vec{Z}\rangle t'''} v^{\langle\vec{\alpha},\vec{Z}\rangle t'} g_2^{\sigma\langle\vec{\alpha},\vec{Z}\rangle t'}, T^{\langle\vec{\alpha},\vec{Z}\rangle t'''} h^{\langle\vec{\alpha},\vec{Z}\rangle t'} g_2^{\langle\vec{\alpha},\vec{Z}\rangle t'}, T^{y_u t'''} g^{y_u t'} g_2^{t''}\}.$$

Identically with lemma 6.2, if $T \in G_{p_1p_2}$, this properly simulates $O_{2.4}$. Also, if $T \in G$, G_{p_3} part of the challenge ciphertext distributed randomly, and this properly simulates $O_{2.3}$. □

Lemma 6.5: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.4}Adv_{\mathcal{A}} - O_{2.5}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given g_1, g_2, X_1X_3, T. It will simulate $O_{2.4}$ or $O_{2.5}$ using $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, \tilde{y} \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1^{y_u}$, $w = g_1^{y_w}$, $v = g_1^{y_v}$ and $h = g_1$. It sends the following group elements to $\mathcal{A}$:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a, (X_1X_3)^{y_w} g_2^{y_w\tilde{y}}, (X_1X_3) g_2^{\tilde{y}}, (X_1X_3)^{y_v} g_2^{y_v\tilde{y}} : j \in [1, \ell]).$$

This implicitly sets $g_1^y = X_1$ modulo p_1 and $g_3^{\tilde{y}}$ modulo p_3. Also, $\psi = y_w$ and $\sigma = y_v$ modulo p_2 and p_3.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y'', r'', \gamma_0'', \ldots \gamma_\ell'' \in \mathbb{Z}_N$ and returns

$$\{(X_1X_3g_2)^{y_w y''}, (X_1X_3g_2)^{y''}, (X_1X_3g_2)^{y_v y''} (X_1X_3)^{r'' y_u} g_2^{\gamma_0''}, (X_1X_3)^{r''(-\alpha_0 x_j/x_0 + \alpha_j)} g_2^{\gamma_j''} : j \in [1, \ell]\}$$

Let us write X_1X_3 as $g_1^{y_{x_1}} g_3^{y_{x_3}}$, this implicitly sets $y' = y_{x_1} y''$ and $r' = y_{x_1} r''$ modulo p_1. $\tilde{y}'$ equals to y'' modulo p_2 and $y_{x_3} y''$ modulo p_3. Also, $\psi = y_w$ modulo p_2 and p_3, and $\sigma = y_v$ modulo p_2 and p_3. γ_0' equals γ_0'' modulo p_2 and $y_{x_3} r'' y_u$ modulo p_3. For $j \in [1, \ell]$, γ_j' equals γ_j'' modulo p_2 and $y_{x_3} r''(-\alpha_0 x_j/x_0 + \alpha_j)$ modulo p_3.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $b, t, \alpha_0', \ldots, \alpha_\ell', t_1, t_2 \in \mathbb{Z}_N$ and returning

$$\{w^s g_2^b T^{y_v\langle\vec{\alpha},\vec{Z}\rangle} g_2^{y_v\langle\vec{\alpha}',\vec{Z}\rangle t_1}, T^{\langle\vec{\alpha},\vec{Z}\rangle} g_2^{\langle\vec{\alpha}',\vec{Z}\rangle t_1}, T^{y_u} g_2^{t_2}\}$$

where $\vec{\alpha}' = (\alpha_0', \ldots, \alpha_\ell')$.

If $T \in G_{p_1}$ and g_1^t is the G_{p_1} part of T, then this matches the distribution of $O_{2.4}$. If $T \in G_{p_1p_3}$, $g_1^t g_3^{t'}$ is the $G_{p_1p_3}$ part of T, this implicitly sets $t_3 = \langle\alpha, \vec{Z}\rangle t'$ modulo p_3 and $t_4 = y_u t'$ modulo p_3. This matches the distribution of $O_{2.5}$ because the G_{p_3} part in the challenge ciphertext is k-wise independent as in Lemma 5.1. □

Lemma 6.6: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.5}Adv_{\mathcal{A}} - O_{2.6}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: $\mathcal{B}$ is given $g_1, g_3, X_1X_2, Y_2Y_3, T$. It will simulate $O_{2.5}$ or $O_{2.6}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, \psi, y \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}$, $w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It sends to $\mathcal{A}$ the group elements:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, X_1X_2, w^y(Y_2Y_3)^\psi, h^y(Y_2Y_3), v^y(Y_2Y_3)^{y_v} : j \in [1, \ell])$$

This is properly distributed if we set $X_1X_2 = g_1^s g_2^a$. Moreover, this implies that $\sigma = y_v$ modulo p_2 and p_3.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y', r', \gamma_0', \ldots, \gamma_\ell' \in \mathbb{Z}_N$ and returns

$$\{w^{y'}(Y_2Y_3)^{\psi y'},\ h^{y'}(Y_2Y_3)^{y'}, v^{y'}u^{r'}(Y_2Y_3)^{\gamma_0'+y_v y'}, h^{r'(-\alpha_0 x_j/x_0+\alpha_j)}(Y_2Y_3)^{\gamma_j'} : j \in [1,\ell]\}.$$

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ returning

$$\{(X_1X_2)^{y_w} T^{y_v\langle\vec{\alpha},\vec{Z}\rangle},\ T^{\langle\vec{\alpha},\vec{Z}\rangle},\ T^{y_u}\}$$

Because y_w and y_v modulo p_2 do not appear anywhere else, $g_2^b = g_2^{ay_w}$ is randomly distributed. $T \in G_{p_1p_3}$, $\langle\vec{\alpha}, \vec{Z}\rangle$ modulo p_3 appears to be uniformly random to the adversary since α_j and y_u modulo p_3 do not appear anywhere else. Hence, this matches the distribution of $O_{2.6}$. If $T \in G$, this implies that $\alpha_j' = \alpha_j$ modulo p_2, $t_1 = t'$ and $t_2 = y_u t'$ where we denote by G_2 the part of T as $g_2^{t_1}$. It should be noted that y_u modulo p_2 does not appear anywhere else. So, t_2 is also uniformly random to the adversary. Therefore, this matches the distribution of $O_{2.5}$. □

Lemma 6.7: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.6}Adv_{\mathcal{A}} - O_{2.7}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given g_1, g_2, X_1X_3, T. It will simulate $O_{2.6}$ or $O_{2.7}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, \tilde{y} \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}$, $u = g_1, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It sends to $\mathcal{A}$ the group elements:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a, (X_1X_3)^{y_w} g_2^{y_w\tilde{y}}, (X_1X_3)^{y_h} g_2^{\tilde{y}}, (X_1X_3)^{y_v} g_2^{y_v\tilde{y}} : j \in [1,\ell]).$$

This is properly distributed. Also, $\psi = y_w$ and $\sigma = y_v$ modulo p_2, p_3.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y'', r', \gamma_0', \ldots, \gamma_\ell' \in \mathbb{Z}_N$ returns

$$\{(X_1X_3g_2)^{y_w y'}, (X_1X_3g_2)^{y'}, (X_1X_3g_2)^{y_v y'}(X_1X_3)^{y_u r'} g_2^{\gamma_0'}, (X_1X_3)^{r'(-\alpha_0 x_j/x_0+\alpha_j)} g_2^{\gamma_j'} : j \in [1, \ell-1]\}$$

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ randomly choose b, t_1, t_2 and responds to $\mathcal{A}$ by returning

$$\{w^s g_2^b T^{y_v\langle\vec{\alpha},\vec{Z}\rangle},\ T^{\langle\vec{\alpha},\vec{Z}\rangle},\ T^{y_u}\}.$$

This implicitly sets g_1^t to be the G_{p_1} part of T. If $T \in G_{p_1}$, then this matches the distribution of $O_{2.7}$. If $T \in G_{p_1p_3}$, for the same reasons as Lemma 6.5, this matches the distribution of $O_{2.6}$. □

Lemma 6.8: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.7}Adv_{\mathcal{A}} - O_{2.8}Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Proof: $\mathcal{B}$ is given $g_1, g_3, X_1X_2, Y_2Y_3, T$. It will simulate $O_{2.7}$ or $O_{2.8}$ with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, y, \psi, \sigma \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}, w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It sends to $\mathcal{A}$ the group elements:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, X_1X_2, w^y(Y_2Y_3)^{\psi y}, h^y(Y_2Y_3)^y, v^y(Y_2Y_3)^{\sigma y} : j \in [0,\ell])$$

When $\mathcal{A}$ makes a ciphertext-type query for the set of recivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $t \in \mathbb{Z}_N$ and returning

$$\{(X_1X_2)^{y_w} g_1^{y_v\langle\vec{\alpha},\vec{Z}\rangle t},\ g_1^{\langle\vec{\alpha},\vec{Z}\rangle t},\ g_1^{y_u t}\}.$$

Then the G_{p_1} part of challenge ciphertext properly distributed if we denotes $X_1X_2 = g_1^s g_1^{y_{x_2}}$. Also, the G_{p_2} part of challenge ciphertext, $b = y_{x_2}y_w$ modulo p_2. This is a properly distributed ciphertext because y_{x_2} modulo p_2 does not appear anywhere else.

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $y'', r' \in \mathbb{Z}_N$ and returns

$$\{w^y(Y_2Y_3)^{\psi y''},\ h^y(Y_2Y_3)^{y''},\ v^y(Y_2Y_3)^{\sigma y''}T^{y_u}, T^{(-a_0x_j/x_0+a_j)} : j \in [1,\ell]\}.$$

The G_{p_1} part of the challenge key is properly distributed if we implicitly set the G_{p_1} part of T as $g_1^{r'}$. Moreover, if we write Y_2Y_3 as $gfrft_2^{y_{y_2}}g_3^{y_{y_3}}$, $\tilde{y}'$ is equal to $y''y_{y_2}$ modulo p_2 and $y''y_{y_3}$ modulo p_3.

If $T \in G_{p_1p_3}$, the challenge key type response is identically distributed to a response from $O_{2.8}$ because α_j and y_u modulo p_3 do not appear anywhere else. If $T \in G$, then the challenge key-type response is identically distributed with a response from $O_{2.7}$. because α_j and y_u modulo p_2 and p_3 do not appear anywhere else. □

Lemma 6.9: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $O_{2.8}Adv_{\mathcal{A}} - O_3Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Proof: $\mathcal{B}$ is given g_1, g_2, X_1X_3, T. It will simulate $O_{2.8}$ or O_3 with $\mathcal{A}$. It chooses random exponents $y_g, y_u, y_w, y_v, \alpha_0, \ldots, \alpha_\ell, s, a, y_2 \in \mathbb{Z}_N$, and sets $g = g_1^{y_g}, u = g_1^{y_u}$, $w = g_1^{y_w}, v = g_1^{y_v}, h = g_1$. It sends to $\mathcal{A}$ the group elements:

$$(g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a, (X_1X_3)^{y_w} g_2^{y_w y_2}, (X_1X_3)g_2^{y_2}, (X_1X_3)^{y_v} g_2^{y_v y_2} : j \in [1,\ell]).$$

This is properly distributed and implies that $\psi = y_w$ and $\sigma = y_v$ modulo p_2, p_3.

When $\mathcal{A}$ makes a ciphertext-type query for the set of receivers, S, $\mathcal{B}$ responds to $\mathcal{A}$ by choosing random $t, t_1, t_2 \in \mathbb{Z}_N$ and returns

$$\{w^s g_2^b v^{\langle\vec{\alpha},\vec{Z}\rangle t},\ h^{\langle\vec{\alpha},\vec{Z}\rangle t},\ u^t\}.$$

When $\mathcal{A}$ makes a challenge key-type query for some identity i, $\mathcal{A}$ chooses a random $\tilde{y}' \in \mathbb{Z}_N$ and returns

$$\{(X_1X_3g_2^{\tilde{y}'})^{y_w},\ (X_1X_3g_2^{\tilde{y}'}), (X_1X_3g_2^{\tilde{y}'})^{y_v}T^{y_u r'},\ T^{r'(-a_0x_j/x_0+a_j)} : j \in [1,\ell]\}.$$

This implicitly sets $g_1^{r'}$ to be the G_{p_1} part of T. If $T \in G_{p_1p_3}$, the G_{p_3} part of the challenge key is properly distributed

because y_u and a_j modulo p_3 do not appear anywhere else. Hence, this matches the distribution of $O_{2.8}$ If $T \in G_{p_1}$, then, this matches the distribution of O_3. □

Lemma 7: Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $Game_{k-1}^{IBBE'} Adv_{\mathcal{A}} - Game_k^{IBBE'} Adv_{\mathcal{A}} = \epsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 2 or Assumption 3.

Proof: We assume there exists a PPT attacker $\mathcal{A}$ who distinguishes between $Game_{k-1}^{IBBE'}$ and $Game_k^{IBBE'}$ with non-negligible advantage. This means that $\mathcal{A}$ can distinguish at least one of following games such as $Game_{k-1}^{IBBE'}$ and $Game_k^{EK}$, $Game_k^{EK}$ and $Game_k^{EC}$, and $Game_k^{EC}$ and $Game_k^{IBBE'}$ with non-negligible advantage. If this adversary exists, this can be used to create a PPT algorithm $\mathcal{B}$ distinguishing one of following pairs of oracles such as O_0 and O_1, O_1 and O_2 and O_2 and O_3 with non-negligible advantage. However, this violates one of Lemmas 4, 5 and 6.

Assuming that $\mathcal{B}$ interacts with one of O_0, O_1, O_2 and O_3. Each oracle outputs as an initial response the group elements

$$\{g, u, w, h, v^{\alpha_j}, h^{\alpha_j}, h^s g_2^a, \\ w^y (g_2 g_3)^{\psi \tilde{y}},\ h^y (g_2 g_3)^{\tilde{y}}, v^y (g_2 g_3)^{\sigma \tilde{y}} : \forall j \in [0, \ell]\}.$$

$\mathcal{B}$ randomly chooses $\delta \in \mathbb{Z}_N$, and gives to $\mathcal{A}$ the public parameters,

$$PK = \{N, G, g, u, w, v^{\alpha_j}, h^{\alpha_j}, e(g,h)^{\delta} : \forall j \in [0, \ell]\}.$$

To create the first $k-1$ semi-functional keys, B generates K_0, K_1, K_2, and $K_{3,j}$ using the key generation algorithm. Then, it randomly chooses $\delta, y_i' \in \mathbb{Z}_N$ and, by using the semi-functional elements in the initial response, constructs semi-functionl keys as:

$$K_0' = g^{\delta} K_0 (w^y (g_2 g_3)^{\psi \tilde{y}})^{y_i'}, K_1' = K_1 (h^y (g_2 g_3)^{\psi \tilde{y}})^{y_i'}, \\ K_2' = K_2 (v^y (g_2 g_3)^{\sigma \tilde{y}})^{y_i'}, K_{3,j}' = K_{3,j} : j \in [1, \ell]$$

This implicitly sets $y_i = y y_i' + y_i''$ modulo p_1 and $y = y y_i'$ modulo p_2, p_3 when we let y_i'' be a randomization parameter shared in the first three components of the normal key for identity i.

For responding normal keys ($> k$), $\mathcal{B}$ generates normal keys by the key generation algorithm. This is possible because $\mathcal{B}$ knows $MSK = \{\delta\}$. It forwarded a normal key to the $\mathcal{A}$.

If $\mathcal{A}$ requests the kth key for some identity i, $\mathcal{B}$ makes a challenge key-type query to the oracle with i. Then, oracle returns group elements, $\{T_0, T_1, T_2, T_{3,j} : j \in [1, \ell]\}$. $\mathcal{B}$ constructs the challenge key for $\mathcal{A}$ as:

$$K_0 = g^{\delta} T_0, K_1 = T_1, K_2 = T_2, K_{3,j} = T_{3,j} : j \in [1, \ell]$$

If the oracle which $\mathcal{B}$ interacts with is O_0, this challenge key is a properly distributed normal key. If the oracle is O_1, this key will be a properly distributed ephemeral key. If the oracle is O_2, this key will be distributed as ephemeral key, properly. If $\mathcal{B}$ is interacting with O_3, this will be distributed as a proper semi-functional key.

When $\mathcal{A}$ requests challenge-ciphertext with the set of receivers S for messages M_0, M_1, $\mathcal{B}$ forwards this query to the oracle and the received group elements (T_1', T_2', T_3'). Then $\mathcal{B}$ choose $f \in \{0, 1\}$, and construct the ciphertext as:

$$C = M_f e(g^{\delta}, h^s g_2^a), C_0 = h^s g_2^a, C_1 = T_1', C_2 = T_2',\ C_3 = T_3'$$

and returns it to $\mathcal{A}$.

If $\mathcal{B}$ is interacting with O_0, O_1, O_3 then the challenge ciphertext will be a properly distributed semi-functional ciphertext. Otherwise, if the oracle which $\mathcal{B}$ interacts is O_2, then the challenge ciphertext will be an properly distributed ephemeral ciphertext.

Thus, if $\mathcal{B}$ interacts with O_0, O_1, O_2 and O_3, then it has properly simulated $Game_{k-1}^{IBBE'}$, $Game_k^{EK}$, $Game_k^{EC}$ and $Game_k^{IBBE'}$, respectively. Thus, if $\mathcal{A}$ distinguishes at least one of the pairs of games with non-negligible advantage, $\mathcal{B}$ can use this to distinguish a corresponding pair of oracles with non-negligible advantage. This violates Lemmas 3, 4 or 5. □

V. Conclusion

In this paper, we introduced the adaptively secure identity-based broadcast encryption scheme featuring constant size ciphertext. The public parameters and private keys in our scheme increase linearly with the maximum number of receivers, but not the total number of users. Also, the computational complexity of the decryption process of our scheme only depends on the number of receivers. Finally, we showed that our scheme is adaptively secure under the general decisional subgroup assumption instead of multiple subgroup decisional assumptions in the standard model through the use of the dual system encryption technique.

References

[1] A. Fiat and M. Naor, "Broadcast encryption," in *Proc. CRYPTO*, 1993, pp. 480–491.

[2] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," in *Proc. 28th Annu. Int. Conf. EUROCRYPT,* 2009, pp. 171–188.

[3] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. CRYPTO*, 2005, pp. 258–275.

[4] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Proc. ASIACRYPT*, 2007, pp. 200–215.

[5] R. Sakai and J. Furukawa, "Identity-based broadcast encryption," *IACR Cryptol. ePrint Archive*, vol. 2007, p. 217, 2007.

[6] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO*, 1984, pp. 47–53.

[7] J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *Proc. 8th Int. Workshop Theory Public Key Cryptography*, 2005, pp. 380–397.

[8] M. Barbosa and P. Farshim, "Efficient identity-based key encapsulation to multiple parties," in *Proc. IMA Int. Conf.*, 2005, pp. 428–441.

[9] N. P. Smart, "Efficient key encapsulation to multiple parties," in *Proc. 4th Int. Conf. SCN*, 2004, pp. 208–219.

[10] Y. Dodis and N. Fazio, "Public key broadcast encryption for stateless receivers," in *Proc. Digit. Rights Manage. Workshop*, 2002, pp. 61–80.

[11] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," in *Proc. 22nd Annu. Int. CRYPTO*, 2002, pp. 47–60.

[12] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proc. 21st Annu. Int. CRYPTO*, 2001, pp. 41–62.

[13] B. Waters, "Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 5677, S. Halevi, Ed. Berlin, Germany: Springer-Verlag, 2009, pp. 619–636.

[14] A. Lewko and B. Waters, "New techniques for dual system encryption and fully secure HIBE with short ciphertexts," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 5978, D. Micciancio, Ed. Berlin, Germany: Springer-Verlag, 2010, pp. 455–479.
[15] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 6110, H. Gilbert, Ed. Berlin, Germany: Springer-Verlag, 2010, pp. 62–91.
[16] M. Bellare, B. Waters, and S. Yilek, "Identity-based encryption secure against selective opening attack," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 6597, Y. Ishai, Ed. Berlin, Germany: Springer-Verlag, 2011, pp. 235–252.
[17] B. Malek and A. Miri, "Adaptively secure broadcast encryption with short ciphertexts," *IJ Netw. Secur.*, vol. 14, no. 2, pp. 71–79, 2012.
[18] Y. Ren and D. Gu, "Fully CCA2 secure identity based broadcast encryption without random oracles," *Inf. Process. Lett.*, vol. 109, no. 11, pp. 527–533, May 2009.
[19] A. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 273–285.
[20] M. Naor and B. Pinkas, "Efficient trace and revoke schemes," in *Proc. 4th Int. Conf. Financial Cryptography*, 2000, pp. 1–20.
[21] N. Attrapadung, "Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 8441, P. Q. Nguyen and E. Oswald, Eds. Berlin, Germany: Springer-Verlag, 2014, pp. 557–577.
[22] D. Boneh, B. Waters, and M. Zhandry, "Low overhead broadcast encryption from multilinear maps," *IACR Cryptol. ePrint Archive*, vol. 2014, p. 195, 2014.
[23] N. Attrapadung and B. Libert, "Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation," in *Proc. 13th Int. Conf. Pract. Theory Public Key Cryptography (PKC)*, vol. 6056. Paris, France, May 2010, pp. 384–402. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13013-7
[24] D. Boneh and M. Hamburg, "Generalized identity based and broadcast encryption schemes," in *Proc. 14th Int. Conf. Theory Appl. Cryptol. Inf. Secur. Adv. Cryptol. (ASIACRYPT)*, vol. 5350. Melbourne, Vic., Australia, Dec. 2008, pp. 455–470.
[25] M. Zhang, B. Yang, Z. Chen, and T. Takagi, "Efficient and adaptively secure broadcast encryption systems," *Secur. Commun. Netw.*, vol. 6, no. 8, pp. 1044–1052, Aug. 2013.
[26] L. Zhang, Y. Hu, and Q. Wu, "Adaptively secure identity-based broadcast encryption with constant size private keys and ciphertexts from the subgroups," *Math. Comput. Model.*, vol. 55, nos. 1–2, pp. 12–18, Jan. 2012.
[27] A. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 6632, K. G. Paterson, Ed. Berlin, Germany: Springer-Verlag, 2011, pp. 547–567.
[28] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 3378, J. Kilian, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 325–341.
[29] A. Guillevic, "Comparing the pairing efficiency over composite-order and prime-order elliptic curves," in *Applied Cryptography and Network Security* (Lecture Notes in Computer Science), vol. 7954, M. Jacobson, Jr., M. Locasto, P. Mohassel, and R. Safavi-Naini, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 357–372.
[30] A. K. Lenstra, "Unbelievable security matching AES security using public key systems," in *Proc. 7th Int. Conf. Theory Appl. Cryptol. Inf. Secur. Adv. Cryptol. (ASIACRYPT)*, Gold Coast, Qld, Australia, Dec. 2001, pp. 67–86.

Jongkil Kim is currently pursuing the Ph.D. degree with the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW, Australia. He is a member of the Centre for Computer and Information Security Research. His main research interest is in functional encryption, including broadcast encryption.

Willy Susilo (SM'01) received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, NSW, Australia. He is currently a Professor with the School of Computer Science and Software Engineering and the Director of the Centre for Computer and Information Security Research, University of Wollongong. He has received the prestigious Australian Research Council Future Fellowship. His main research interests include cryptography and information security. He has authored numerous publications in the area of digital signature schemes and encryption schemes.

Man Ho Au (M'12) is currently an Assistant Professor with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His research interests include information security and privacy. He has authored over 60 referred journal and conference papers, including two papers in the ACM Conference on Computer and Communications Security that were named as the Runners-Up for the Pet Award 2009: Outstanding Research in Privacy Enhancing Technologies.

Jennifer Seberry (SM'97) received the Ph.D. degree in computation mathematics from La Trobe University, Melbourne VIC, Australia, in 1971. She is currently a Professor with the School of Computer Science and Software Engineering and the Founding Director of the Centre for Computer Security Research, University of Wollongong, Wollongong, NSW, Australia. She has published extensively in Discrete Mathematics and is world renown for her new discoveries on Hadamard matrices, orthogonal designs, and statistical design. Because of her outstanding contribution to cryptologic research, she has been a fellow of the International Association for Cryptologic Research since 2012.