Fully Secure Anonymous Hierarchical Identity-Based Encryption with Constant Size Ciphertexts

Jae Hong Seo and Jung Hee Cheon

Department of Mathematical Sciences and ISac-RIM, Seoul National University, Seoul, 151-747, Korea
{jhsbhs0,jhcheon}@snu.ac.kr

Abstract. Efficient and privacy-preserving constructions for search functionality on encrypted data is important issues for data outsourcing, and data retrieval, etc. Fully secure anonymous Hierarchical ID-Based Encryption (HIBE) schemes is useful primitives that can be applicable to searchable encryptions [4], such as *ID-based searchable encryption*, *temporary searchable encryption* [1], and *anonymous forward secure HIBE* [9]. We propose a fully secure anonymous HIBE scheme with constant size ciphertexts.

keywords. Hierarchical Identity-Based Encryption (HIBE), Anonymous Hierarchical Identity-Based Encryption, Public-key Encryption with Keyword Search (PEKS)

1 Introduction

Shamir introduced the notion of Identity-Based Encryption (IBE) which is a public-key cryptosystem being able to use any string, such as e-mail, as a public key [20], and Boneh and Franklin proposed the first IBE scheme using pairing [5]. Hierarchical Identity Based Encryption (HIBE) is an extension allowing high level users to delegate their key generation ability to the low level users [16, 15]. Abdalla et. al. introduced the notion of anonymous IBE and anonymous HIBE that satisfy an additional privacy requirement such that no adversary can obtain information for the recipient's identity ID from ciphertexts if she do not have a private key of ID or its ancestors' [1]. Both anonymous IBE and anonymous HIBE is useful primitives that can be applicable to encryption systems allowing search functionality on encrypted data [4, 1, 9, 7, 21].

The first realization of an anonymous HIBE scheme is proposed by Boyen and Waters [9]. Since Boyen and Waters' anonymous HIBE, several approaches to build an anonymous HIBE scheme are introduced [22, 19, 14]. However, all previous anonymous HIBE schemes proved their securities in the *selective* security model that restricts the adversary to commit the target ID before that public parameters are generated by the challenger in the security game. *Selective* security notion does not reflect real adversaries' behaviors sufficiently. In contrast to *selective* security model, *full* security model allows the adversary to be able to choose the target identity after obtaining public system parameters and private keys which are adaptively chosen by the adversary. Therefore, *full* security is stronger security notion than *selective* security, and it reflects real world adversary well. We propose a HIBE scheme satisfying following properties together, *full* security, anonymity, and constant size ciphertexts.

Our Contributions: Our construction is inspired by two HIBE schemes proposed by Lewko and Waters [18], and Seo et. al. [19]. The HIBE scheme in [18] achieves full security and constant size ciphertexts, but not anonymity. On the other hand, Seo et. al.'s HIBE scheme attains anonymity and constant size ciphertexts, but not full security. We note that our construction is not a simple combination of two schemes. Let us explain what is a hard task if we

combine techniques in two schemes. Seo et. al.'s ideas to obtain anonymity are blinding public parameters and ciphertexts, and adding re-randomization subkeys into private keys. In their scheme, adding re-randomization subkeys into private keys does not impact the security proof since re-randomization subkeys do not contain the master secret key used to decrypt. More precisely, in the security proof of *selective* model, the simulator know the target ID^* before he generates public parameters, so that he can generate public parameters to allow to be able to generate all private keys except for the target ID^*. That is, when the simulator generates public parameters, the element hard to compute in the underlying hard problem is embedded to the private key for ID^*. Hence the simulator can generate almost all elements except the private key for target ID^* and its ancestors', and so he can easily generate re-randomization subkeys for all private keys. However, this strategy cannot apply to the full security model directly. Since the simulator cannot see target ID^* before generating public parameters, the simulator should be able to generate all private keys to reply key extraction queries. Therefore, adding re-randomization subkeys to private keys is not an easy work contrast to the scheme in [19].

We construct the scheme in bilinear groups of composite order of four primes, and give the provable security of our construction under six new static assumptions. Even though our construction use composite order group of four primes, we claim that our construction is practical in comparison with other anonymous HIBE schemes. All selective secure (H)IBE scheme can be transfered to the full secure scheme by increasing group size [2]. This transformation increases, however, the group size exponentially according to the maximum hierarchical depth, eventually resulting schemes are very inefficient compare to our construction. Moreover, assumptions used to prove confidentiality and anonymity of our scheme are static (but not standard). I.e. assumptions are independent from the maximum number of the adversary's private key queries.

Applications: Anonymous IBE and HIBE have variety applications in search on encrypted data of public-key cryptosystems, such as Public-key Encryption with Keyword Search (PEKS) [4, 1]. PEKS is a useful primitives for constructing secure audit logs [24, 13], secure multi-dimensional range query [21], conjunctive keyword search [7], and anonymous credential [10]. ID-based searchable encryptions and temporary searchable encryptions are extensions of PEKS. For example, we can use two level anonymous HIBE scheme where the first level is used for user's identities and the second level is used for keywords. This is an combination of IBE and PEKS, called Identity-Based Encryption with Keyword Search (IBEKS) proposed in [1]. In IBEKS scheme, each user in the first level of anonymous HIBE scheme can generate all tokens for keywords chosen by himself using his private key without requiring to a central authority. Public-key Encryption with Temporary Keyword Search (PETKS) is also a useful application of anonymous HIBE [1]. In PETKS scheme, intermediate nodes in hierarchy of anonymous HIBE is corresponds to time periods and leaf nodes are corresponds to keywords. The time travel of PETKS scheme is defined as in forward secure public-key encryption that is an important application of HIBE [11]. Then, users can generate a token for keyword which is available in temporary time periods defined by users. Forward secure public-key encryption [11] and forward-secure HIBE scheme [25] can be constructed using a HIBE scheme as a building block. If we use an anonymous HIBE scheme instead a HIBE scheme, then we can obtain an anonymous forward secure HIBE scheme [9].

2 Definitions

In this section we define anonymous HIBE scheme and give their security models.

2.1 Anonymous HIBE scheme

Every user of HIBE scheme has an ID consisting of a vector as a public key such as $ID = [I_1, \cdots, I_k]$ where k means user's position in the hierarchy. We sometimes denote $ID|_k$ to emphasize the length of ID instead of ID when the length of ID is k. The root node of hierarchy means Private Key Generator (PKG), denoted by $ID|_0$.

Definition 1. *A HIBE scheme consists of four probabilistic algorithms, Setup, KeyGen, Enc and Dec algorithms as follows.*

- *Setup*$(\lambda, \ell) \rightarrow \{params, MSK\}$. *Setup takes the security parameter λ and the maximum hierarchical depth ℓ as input, and it generates public system parameters, denoted by params and the master secret key, denoted by $MSK = Pvk_{ID|_0}$. params includes the message space $\mathcal{M}$, the ciphertext space $\mathcal{CT}$ and the identity space $\mathcal{I}$. MSK is kept by PKG as secret values.*
- *KeyGen*$(Pvk_{ID|_\tau}, ID|_k) \rightarrow \{Pvk_{ID|_k}\}$. *KeyGen generates the private key $Pvk_{ID|_k}$ of the identity $ID|_k$ using the private key $Pvk_{ID|_\tau}$ for the identity $ID|_\tau$ where $\tau < k$ and $ID|_\tau$ is an ancestor identity of $ID|_k$.*
- *Enc*$(params,\ ID,\ M) \rightarrow \{CT\}$. *Enc outputs a ciphertext $CT \in \mathcal{CT}$ for a message $M \in \mathcal{M}$ and a recipient identity $ID \in \mathcal{I}$.*
- *Dec*$(Pvk_{ID}, CT) \rightarrow \{M\}$. *Dec returns the message $M \in \mathcal{M}$.*

Enc and Dec have to satisfy the consistency constraint such that for every identity $ID \in \mathcal{I}$ and the corresponding private key Pvk_{ID} generated by KeyGen and every message $M \in \mathcal{M}$,

$$\mathsf{Dec}(Pvk_{ID}, \mathsf{Enc}(params, ID, M)) = M$$

where the probability goes over all randomness used in all algorithms above.

2.2 Security Models

We deal with two kinds of security notions, the confidentiality and the anonymity. Confidentiality means that ciphertexts does not leak information about corresponding plaintexts, and the anonymity means recipient's privacy. Both of security notions are defined by games between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, IND-ID-CPA game for confidentiality and ANON-ID-CPA game for anonymity.

IND-ID-CPA Game:

- Setup. $\mathcal{C}$ runs Setup and gives $\mathcal{A}$ public system parameters and retains the master secret key as secret values.
- Query Phase 1. $\mathcal{A}$ adaptively issues identities ID. $\mathcal{C}$ generates Pvk_{ID} by running KeyGen, and sends Pvk_{ID} to $\mathcal{A}$.
- Challenge. $\mathcal{A}$ outputs two equal length messages M_0, M_1 and a target identity ID^*. The target identity ID^* and its prefixes have not queried before. Then, $\mathcal{C}$ flips a random coin β and makes the challenge ciphertext, Enc($params$, ID^*, M_β). Then sends it to the adversary.

Query Phase 2. Repeat Query Phase 1. The only restriction is $\mathcal{A}$ cannot query for the target identity ID^* and its prefixes.
Guess. $\mathcal{A}$ outputs a guess β' of β, and then wins if $\beta = \beta'$.

The advantage of $\mathcal{A}$ in the above game is defined as the absolute value of the difference between the probability of $\beta = \beta'$ and $1/2$.

Definition 2. *We say that an HIBE scheme is IND-ID-CPA secure if for any polynomial time adversary, its advantage in IND-ID-CPA game is negligible.*

ANON-ID-CPA Game:

Setup. $\mathcal{C}$ runs Setup and gives $\mathcal{A}$ the public system parameters and retains the master secret key as secret values.
Query Phase 1. $\mathcal{A}$ adaptively issues identities ID. $\mathcal{C}$ generates Pvk_{ID} by running KeyGen, and sends Pvk_{ID} to $\mathcal{A}$.
Challenge. $\mathcal{A}$ outputs message M and two target identities ID_0^* and ID_1^*. Both of two target identities and their prefixes have not queried before. Then, $\mathcal{C}$ flips a random coin β and makes the challenge ciphertext, Enc($params$, ID_β^*, M). Then sends it to the adversary.
Query Phase 2. Repeat Query Phase 1. The only restriction is $\mathcal{A}$ cannot query for the target identities and their prefixes.
Guess. $\mathcal{A}$ outputs a guess β' of β, and then wins if $\beta = \beta'$.

The advantage of $\mathcal{A}$ in ANON-ID-CPA game is defined as the absolute value of the difference between the probability of $\beta = \beta'$ and $1/2$.

Definition 3. *We say that an HIBE scheme is ANON-ID-CPA secure if for any polynomial time adversary, its advantage in ANON-ID-CPA game is negligible.*

We can extend the above security notions to the CCA security notions, IND-ID-CCA and ANON-ID-CCA by allowing the adversary to use the decryption oracle in Query Phases of both games. CCA security can be achieved from CPA security by using techniques that are method of transforming from CPA-secure $(\ell+1)$-level HIBE to CCA-secure ℓ-level HIBE, for example [3, 8]. Therefore in this paper we only focus on CPA security notions.

3 Background in Mathematics and Complexity Assumptions

3.1 Bilinear Groups of Composite Order

We will use a bilinear group of composite order $n = p_1p_2p_3p_4$. Bilinear groups of composite order were introduced by Boneh, Goh, and Nissim [6]. Many literatures make cryptographic schemes over composite order bilinear groups [6, 7, 17, 22, 23, 19, 18].

Let $\mathcal{G}$ be a group generating algorithm that takes a security parameter λ as a input and outputs a tuple

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$$

where p_1, p_2, p_3 and p_4 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $n = p_1p_2p_3p_4$, and e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear map; i.e., e satisfies the following properties:

1. (bilinear) For $\forall g_1, h_1 \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, $e(g_1^a, h_1^b) = e(g_1, h_1)^{ab}$.
2. (non-degenerate) For generator g_1 of $\mathbb{G}$, $e(g_1, g_1)$ generates $\mathbb{G}_T$.

3. (efficiently computable) There exists an efficient algorithm that computes bilinear map e in polynomial time with respect to λ.

We assume that group operations in $\mathbb{G}$ and $\mathbb{G}_T$ are all computable in polynomial time with respect to λ. Furthermore, we assume that descriptions of $\mathbb{G}$ and $\mathbb{G}_T$ contain generators as well as identity elements $1_{\mathbb{G}}$, $1_{\mathbb{G}_T}$ of $\mathbb{G}$ and $\mathbb{G}_T$, respectively.

We will use a notation $\mathbb{G}_{p_i}$ to denote a subgroup of $\mathbb{G}$ of order p_i. Then $\mathbb{G}$ is a direct product of $\mathbb{G}_{p_i}$'s, $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$. We use notations $\mathbb{G}_{p_ip_j}$ and $\mathbb{G}_{p_ip_jp_k}$ to denote subgroups of order p_ip_j and $p_ip_jp_k$, respectively. Since $\mathbb{G}$ has a composite order $n = p_1p_2p_3p_4$, subgroups with order as a factor of N exist, hence such notations make sense.

If X is a generator of $\mathbb{G}$, then $X^{p_2p_3p_4}$, denote to X_1, is a generator of $\mathbb{G}_{p_1}$. Similarly $X^{p_1p_3p_4}$, $X^{p_1p_2p_4}$, $X^{p_1p_2p_3}$ are generators of $\mathbb{G}_{p_2}$, $\mathbb{G}_{p_3}$, $\mathbb{G}_{p_4}$, respectively, and denote to X_2, X_3, X_4, respectively. We note that $e(R_i, R_j) = 1$ for distinct i and j, and all random elements $R_i \in \mathbb{G}_{p_i}$, $R_j \in \mathbb{G}_{p_j}$. This is followed from the fact that $e(R_i, R_j) = e(X_i^a, X_j^b)$ for some integers $a, b \in \mathbb{Z}_N$, and $e(X_i^a, X_j^b) = e(X^{\frac{p_1p_2p_3p_4}{p_i}a}, X^{\frac{p_1p_2p_3p_4}{p_j}b}) = e(X, X)^{p_1p_2p_3p_4\frac{p_1p_2p_3p_4}{p_ip_j}ab} = 1$ since $i \neq j$.

3.2 Complexity Assumptions

We need six complexity assumptions to prove the security of our anonymous HIBE construction. Our assumptions are not standard assumptions, however, these guarantee the security against adversarial strategy that does not use the properties of group representation if the finding nontrivial factors of the group order is hard. The hardness of our assumptions relies on the theorems of Katz, Sahai, and Waters [17].

Assumption 1:
For a given group generator $\mathcal{G}$, let the following distribution be $P_1(\lambda)$.

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda), \quad n \leftarrow p_1p_2p_3p_4,$$

$$g \xleftarrow{R} \mathbb{G}_{p_1}, \quad X_3 \xleftarrow{R} \mathbb{G}_{p_3}, \quad X_4 \xleftarrow{R} \mathbb{G}_{p_4}$$

$$D \leftarrow (\mathbb{G}, n, g, X_3, X_4), \quad T_0 \xleftarrow{R} \mathbb{G}_{p_1p_2p_4}, \quad T_1 \xleftarrow{R} \mathbb{G}_{p_1p_4},$$

$$\beta \xleftarrow{R} \{0, 1\}, \quad T \leftarrow T_0 \cdot (1 - \beta) + T_1 \cdot \beta.$$

Give (D, T) to the adversary $\mathcal{B}$. Then $\mathcal{B}$ outputs β', and succeeds if $\beta = \beta'$. We define the advantage of the adversary $\mathcal{B}$ above, denote to $Adv1_{\mathcal{G},\mathcal{B}}(\lambda)$, in group generated by $\mathcal{G}$ to be the absolute value of the difference of the success probability of the adversary and $1/2$, where the probability is over the distribution $P_1(\lambda)$ and the random coins of $\mathcal{B}$.

Definition 4. *We say that a group generator $\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{G},\mathcal{B}}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{B}$.*

Assumption 2:
For a given group generator $\mathcal{G}$, let the following distribution be $P_2(\lambda)$.

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda), \quad n \leftarrow p_1p_2p_3p_4,$$

$$g, X_1 \xleftarrow{R} \mathbb{G}_{p_1}, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3, Y_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4}$$

$$D \leftarrow (\mathbb{G}, n, g, X_1X_2, X_3, Y_2Y_3, X_4), \quad T_0 \xleftarrow{R} \mathbb{G}_{p_1p_2p_3}, \quad T_1 \xleftarrow{R} \mathbb{G}_{p_1p_3},$$

$$\beta \xleftarrow{R} \{0,1\}, \quad T \leftarrow T_0 \cdot (1-\beta) + T_1 \cdot \beta.$$

Give (D, T) to the adversary $\mathcal{B}$. Then $\mathcal{B}$ outputs β', and succeeds if $\beta = \beta'$. We define the advantage of the adversary $\mathcal{B}$ above, denote to $Adv2_{\mathcal{G},\mathcal{B}}(\lambda)$, in group generated by $\mathcal{G}$ to be the absolute value of the difference of the success probability of $\mathcal{B}$ and $1/2$, where the probability is over the distribution $P_2(\lambda)$ and the random coins of $\mathcal{B}$.

Definition 5. *We say that a group generator $\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{G},\mathcal{B}}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{B}$.*

Assumption 3:

For a given group generator $\mathcal{G}$, let the following distribution be $P_3(\lambda)$.

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda), \quad n \leftarrow p_1 p_2 p_3 p_4,$$

$$X_1 \xleftarrow{R} \mathbb{G}_{p_1}, \quad Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, \quad X_3, Y_3, Y_3' \xleftarrow{R} \mathbb{G}_{p_3}, \quad X_4 \xleftarrow{R} \mathbb{G}_{p_4}$$
$$D \leftarrow (\mathbb{G}, n, X_1, Y_2 Y_3, X_3, X_4), \quad T_0 \leftarrow Y_2 Y_3', \quad T_1 \xleftarrow{R} \mathbb{G}_{p_2 p_3},$$
$$\beta \xleftarrow{R} \{0,1\}, \quad T \leftarrow T_0 \cdot (1-\beta) + T_1 \cdot \beta.$$

Give (D, T) to the adversary $\mathcal{B}$. Then $\mathcal{A}$ outputs β', and succeeds if $\beta = \beta'$. We define the advantage of the adversary $\mathcal{B}$ above, denote to $Adv3_{\mathcal{G},\mathcal{B}}(\lambda)$, in groups generated by $\mathcal{G}$ to be the absolute value of the difference of the success probability of $\mathcal{B}$ and $1/2$, where the probability is over the distribution $P_3(\lambda)$ and the random coins of $\mathcal{B}$.

Definition 6. *We say that a group generator $\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{G},B}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{B}$.*

Assumption 4:

For a given group generator $\mathcal{G}$, let the following distribution be $P_4(\lambda)$.

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda), \quad n \leftarrow p_1 p_2 p_3 p_4,$$

$$X_1 \xleftarrow{R} \mathbb{G}_{p_1}, \quad Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, \quad X_3 \xleftarrow{R} \mathbb{G}_{p_3}, \quad X_4, Y_4 \xleftarrow{R} \mathbb{G}_{p_4}$$
$$D \leftarrow (\mathbb{G}, n, X_1, Y_2 Y_4, X_3, X_4), \quad T_0 \xleftarrow{R} \mathbb{G}_{p_2 p_4}, \quad T_1 \xleftarrow{R} \mathbb{G}_{p_4},$$
$$\beta \xleftarrow{R} \{0,1\}, \quad T \leftarrow T_0 \cdot (1-\beta) + T_1 \cdot \beta.$$

Give (D, T) to the adversary $\mathcal{B}$. Then $\mathcal{A}$ outputs β', and succeeds if $\beta = \beta'$. We define the advantage of the adversary $\mathcal{B}$ above, denote to $Adv4_{\mathcal{G},\mathcal{B}}(\lambda)$, in groups generated by $\mathcal{G}$ to be the absolute value of the difference of the success probability of $\mathcal{B}$ and $1/2$, where the probability is over the distribution $P_4(\lambda)$ and the random coins of $\mathcal{B}$.

Definition 7. *We say that a group generator $\mathcal{G}$ satisfies Assumption 4 if $Adv4_{\mathcal{G},B}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{B}$.*

Assumption 5:

For a given group generator $\mathcal{G}$, let the following distribution be $P_5(\lambda)$.

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda), \quad n \leftarrow p_1 p_2 p_3 p_4,$$

$$g, X_1, Y_1 \xleftarrow{R} \mathbb{G}_{p_1}, X_2, Y_2, Z_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3, Z_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4}$$
$$D \leftarrow (\mathbb{G}, n, g, X_1 X_2, X_3, Y_1 Y_2, Z_2 Z_3, X_4),$$
$$T_0 \leftarrow e(X_1, Y_1), \quad T_1 \xleftarrow{R} \mathbb{G}_T,$$
$$\beta \xleftarrow{R} \{0,1\}, \quad T \leftarrow T_0 \cdot (1-\beta) + T_1 \cdot \beta.$$

Give (D, T) to the adversary $\mathcal{B}$. Then $\mathcal{B}$ outputs β', and succeeds if $\beta = \beta'$. We define the advantage of the adversary $\mathcal{B}$ above, denote to $Adv5_{\mathcal{G},\mathcal{B}}(\lambda)$, in groups generated by $\mathcal{G}$ to be the absolute value of the difference of the success probability of $\mathcal{B}$ and $1/2$, where the probability is over the distribution $P_5(\lambda)$ and the random coins of $\mathcal{B}$.

Definition 8. *We say that a group generator $\mathcal{G}$ satisfies Assumption 5 if $Adv5_{\mathcal{G},B}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{B}$.*

Assumption 6:

We uses Assumption 6 to prove the anonymity of our anonymous HIBE construction. For a given group generator $\mathcal{G}$, let the following distribution be $P_6(\lambda)$.

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(\lambda), \quad n \leftarrow p_1p_2p_3p_4,$$

$$X_1, Y_1, W_1 \xleftarrow{R} \mathbb{G}_{p_1}, \quad Y_2, Z_2, W_2, W_2' \xleftarrow{R} \mathbb{G}_{p_2}, \quad Z_3 \xleftarrow{R} \mathbb{G}_{p_3},$$

$$X_4, Z_4, W_4, W_4' \xleftarrow{R} \mathbb{G}_{p_4}$$

$$D \leftarrow (\mathbb{G}, n, X_1X_4, Y_1Y_2, Z_2, Z_3, Z_4, W_1W_2W_4),$$

$$T_0 \leftarrow W_1W_2'W_4', \quad T_1 \xleftarrow{R} \mathbb{G}_{p_1p_2p_4},$$

$$\beta \xleftarrow{R} \{0,1\}, \quad T \leftarrow T_0 \cdot (1-\beta) + T_1 \cdot \beta.$$

Give (D, T) to the adversary $\mathcal{B}$. Then $\mathcal{B}$ outputs β', and succeeds if $\beta = \beta'$. We define the advantage of the adversary $\mathcal{B}$ above, denote to $Adv6_{\mathcal{G},\mathcal{B}}(\lambda)$, in groups generated by $\mathcal{G}$ to be the absolute value of the difference of the success probability of $\mathcal{B}$ and $1/2$, where the probability is over the distribution $P_6(\lambda)$ and the random coins of $\mathcal{B}$.

Definition 9. *We say that a group generator $\mathcal{G}$ satisfies Assumption 6 if $Adv6_{\mathcal{G},\mathcal{B}}(\lambda)$ is a negligible function of λ for any polynomial time adversary $\mathcal{B}$.*

4 Construction

In this section we proposed a fully secure anonymous HIBE with constant size ciphertexts. We build a scheme in bilinear groups $\mathbb{G}$ of composite order of product of four primes, $n = p_1p_2p_3p_4$. We utilize subgroups $\mathbb{G}_{p_1}$, $\mathbb{G}_{p_2}$, $\mathbb{G}_{p_3}$, $\mathbb{G}_{p_4}$ of $\mathbb{G}$ for different usage. All meaningful information are embedded in a subgroup $\mathbb{G}_{p_1}$. Subgroups $\mathbb{G}_{p_3}$ and $\mathbb{G}_{p_4}$ are respectively used private keys and public parameters to look like random. Subgroup $\mathbb{G}_{p_2}$ is not appeared in real scheme. We use $\mathbb{G}_{p_2}$ only for security proof.

All public parameters and ciphertexts are blinded by random elements of $\mathbb{G}_{p_4}$, so ciphertexts does not leak ID information. If the private key does not have blinding factors in $\mathbb{G}_{p_4}$, blinding factors of ciphertexts will be removed during paring operation in the decryption procedure. HIBE schemes usually use public parameters to re-randomize children's key in delegation algorithm, however, if public parameters have blinding factors, we cannot use public parameters to re-randomize children's key. If then, decryption algorithm will not work correctly. Therefore we need to add re-randomization subkey to the private key. We now describe our construction with keeping this idea in mind.

$\mathsf{Setup}(\lambda, \ell)$: First, the setup algorithm runs group generator $\mathcal{G}$ and obtains $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$. Next, it chooses random elements $g, h, u_1, \cdots, u_\ell, w, \in \mathbb{G}_{p_1}$, $X_3 \in \mathbb{G}_{p_3}$, $X_4 \in \mathbb{G}_{p_4}$, $R_{4,g}$, $R_{4,h}$, R_{4,u_1}, $\cdots$, $R_{4,u_\ell} \in \mathbb{G}_{p_4}$. It then, sets $n = p_1p_2p_3p_4, G = gR_{4,g}$, $H = hR_{4,h}$, $U_1 = u_1R_{4,u_1}, \cdots, U_\ell = u_\ell R_{4,u_\ell}$ and $E = e(g, w)$, and

$$params \leftarrow [\mathbb{G}, n, G, H, U_1, \cdots, U_\ell, X_3, X_4, E], \quad MSK \leftarrow [g,\ h,\ u_1,\ \cdots,\ u_\ell,\ w]$$

Lastly, it publishes the *params* and retain the MSK as secret values.
$\mathsf{Enc}(params, ID, M)$: Parse ID to $[I_1, \cdots, I_k]$. Enc picks a random integer $s \in \mathbb{Z}_n$ and random elements $\bar{R}_4, \bar{R}'_4 \in \mathbb{G}_{p_4}$. A random element of $\mathbb{G}_{p_4}$ can be chosen by raising X_4 to random exponents from $\mathbb{Z}_N$. Next, it sets

$$CT \leftarrow [\ \ C_0 = ME^s, C_1 = (H\prod_{i=1}^{k} U_i^{I_i})^s \bar{R}_4,\ C_2 = G^s \bar{R}'_4\ \ ] \in \mathbb{G}_T \times \mathbb{G}^3.$$

$\mathsf{KeyGen}(MSK, ID)$: Parse ID to $[I_1, \cdots, I_k]$. KeyGen algorithm picks random integers $r_1, r_2 \in \mathbb{Z}_N$ and random elements

$$R_3^{(d)}, R_3'^{(d)}, R_{k+1}^{(d)}, \cdots, R_\ell^{(d)}, R_3^{(r)}, R_3'^{(r)}, R_{k+1}^{(r)}, \cdots, R_\ell^{(r)} \in \mathbb{G}_{p_3}^{2(\ell-k)+4}.$$

The private key Pvk_{ID} consists of two subkeys $Pvk_{ID}^{(d)} \in \mathbb{G}_{p_1p_3}^{\ell-k+2}$ and $Pvk_{ID}^{(r)} \in \mathbb{G}_{p_1p_3}^{\ell-k+2}$. $Pvk_{ID}^{(d)}$ is used for decryption and delegation, and $Pvk_{ID}^{(r)}$ is used for re-randomization. It sets

$$\begin{aligned} Pvk_{ID}^{(d)} &\leftarrow [K_1^{(d)} = g^{r_1} R_3^{(d)}, K_2^{(d)} = w(h\prod_{i=1}^{k} u_i^{I_i})^{r_1} R_3'^{(d)}, E_{k+1}^{(d)} = u_{k+1}^{r_1} R_{j+1}^{(d)}, \cdots, u_\ell^{r_1} R_\ell^{(d)}\ \ ]. \\ Pvk_{ID}^{(r)} &\leftarrow [K_1^{(r)} = g^{r_2} R_3^{(r)}, K_2^{(r)} = \quad (h\prod_{i=1}^{k} u_i^{I_i})^{r_2} R_3'^{(r)}, E_{k+1}^{(r)} = u_{k+1}^{r_2} R_{j+1}^{(r)}, \cdots, u_\ell^{r_2} R_\ell^{(r)}\ \ ]. \end{aligned}$$

$\mathsf{KeyGen}(Pvk_{ID|_{k-1}}, ID|_k)$: Given a private key $Pvk_{ID|_{k-1}}$ for $2 \le k \le \ell$, this algorithm derives the private key for $ID|_k$. Parse $Pvk_{ID|_{k-1}}$ to $Pvk_{ID|_{k-1}}^{(d)} = [K_1^{(d)}, K_2^{(d)}, E_k^{(d)}, \cdots, E_\ell^{(d)}]$ and $Pvk_{ID|_{k-1}}^{(r)} = [K_1^{(r)}, K_2^{(r)}, E_k^{(r)}, \cdots, E_\ell^{(r)}]$. This algorithm consists of two steps, *Delegate* step and *Re-randomize* step. In *Delegate* step, it generates the private key for child, $ID|_k$. The result of *Delegate* step is sufficient to decrypt the ciphertext for $ID|_k$, however, the randomness of these keys are associated with parents keys. It means that distributions of private keys generated by *Delegate* step are different from private keys generated by MSK. We set two distributions to be same by carrying out *Re-randomize* step after *Delegate* step.

Step 1 (Delegate Step): for $\forall i \in [k+1, \ell]$,

$$[K_1'^{(d)}, K_2'^{(d)}, E_i'^{(d)}] \leftarrow [K_1^{(d)}, K_2^{(d)}(E_k^{(d)})^{I_k}, E_i^{(d)}],$$

$$[K_1'^{(r)}, K_2'^{(r)}, E_i'^{(r)}] \leftarrow [K_1^{(r)}, K_2^{(r)}(E_k^{(r)})^{I_k}, E_i^{(r)}].$$

Step 2 (Re-randomize Step): Choose two random integers $s, t \in \mathbb{Z}_N$ and random elements $\bar{R}_3^{(d)}, \bar{R}_3'^{(d)}, \bar{R}_{k+1}^{(d)}, \cdots, \bar{R}_\ell^{(d)}, \bar{R}_3^{(r)}, \bar{R}_3'^{(r)}, \bar{R}_{k+1}^{(r)}, \cdots, \bar{R}_\ell^{(r)}$ from $\mathbb{G}_{p_3}$. Random elements in $\mathbb{G}_{p_3}$ can be generated by raising X_3 to random exponent from $\mathbb{Z}_n$. $Pvk_{ID|_k}^{(d)}$ and $Pvk_{ID|_k}^{(r)}$ are respectively re-randomized as follows:

$$[K_1'^{(d)}(K_1'^{(r)})^s \bar{R}_3^{(d)}, K_2'^{(d)}(K_2'^{(r)})^s \bar{R}_3'^{(d)}, E_i'^{(d)}(E_i'^{(r)})^s \bar{R}_i^{(d)}],$$

$$[\qquad (K_1'^{(r)})^t \bar{R}_3^{(r)}, \qquad (K_2'^{(r)})^t \bar{R}_3'^{(r)}, \qquad (E_i'^{(r)})^t \bar{R}_i^{(r)}].$$

$\mathsf{Dec}(Pvk_{ID}, CT)$: Parse ID, CT and Pvk_d^{ID} to $[I_1, \cdots, I_k]$, $[C_0, C_1, C_2]$ and $[K_1^{(d)}, K_2^{(d)}, E_{k+1}^{(d)}, E_\ell^{(d)}]$, respectively. Then Dec outputs

$$M \leftarrow C_0 \cdot \frac{e(K_1^{(d)}, C_1)}{e(K_2^{(d)}, C_2)}$$

We can easily check the correctness of the Dec algorithm for a valid ciphertext so that we omit details.

5 Security Analysis

To prove the security of our anonymous HIBE scheme, we take the proof methodology of [23, 18]. In other word, we first define semi-functional ciphertexts and semi-functional keys, and we will show that the real security game is computationally indistinguishable from a game that all query results are semi-functional ones. In the real game, the simulator can always check whether the challenge ciphertext is valid or not by generating the corresponding private key himself. Therefore it is uneasy to make reduction to the hard problem. If the simulator, however, can only generate semi-functional ones (ciphertexts and keys), he cannot check by himself the validity of the ciphertexts since semi-functional keys cannot decrypt the semi-functional ciphertext except for the special case. Therefore it is possible to make reduction to the hard problem. Semi-functional ciphertexts are of the form

$$C_0 = C_0', C_1 = C_1' g_2^{x z_c}, C_2 = C_2' g_2^x$$

where C_0', C_1', C_2' are the result of Enc algorithm, $g_2 \in \mathbb{G}_{p_2}$, and $x, z_c \overset{R}{\leftarrow} \mathbb{Z}_N$. Semi-functional keys are of the form

$$K_1^{(d)} = K_1'^{(d)} g_2^{\gamma}, K_2^{(d)} = K_2'^{(d)} g_2^{\gamma z_k}, E_i^{(d)} = E_i'^{(d)} g_2^{\gamma z_i},$$
$$K_1^{(r)} = K_1'^{(r)} g_2^{\gamma'}, K_2^{(r)} = K_2'^{(r)} g_2^{\gamma' z_k'}, E_i^{(r)} = E_i'^{(r)} g_2^{\gamma' z_i'},$$

for $\forall\, i \in [j+1, \ell]$, where $K_1', K_2', E_{j+1}', \cdots E_\ell'$ are the result of KeyGen algorithm, $g_2 \in \mathbb{G}_{p_2}$, and $\gamma, \gamma', z_k, z_{j+1}, \cdots, z_\ell \overset{R}{\leftarrow} \mathbb{Z}_N$. Since elements of $\mathbb{G}_{p_2}$ are used in the semi-functional ones, Dec algorithm will remove elements of $\mathbb{G}_{p_2}$ if it takes semi-functional keys and normal ciphertexts, or normal keys and semi-functional ciphertexts. However, Dec algorithm outputs the result multiplied by additional term $e(g_2, g_2)^{x\gamma(z_c - z_k)}$ if it takes semi-functional keys and semi-functional ciphertexts. If $z_k = z_c$, then the additional term is $1_{\mathbb{G}_T}$, so that decryption will be correct.

We uses a hybrid argument to prove the confidentiality. The first game is the real IND-ID-CPA game, denote to $Game_{Real}$. The second game $Game_{Restricted}$ restricts that the adversary cannot query for the private key for identities which are prefixes of the challenge identity modulus p_2, and remains others are same to $Game_{Real}$. Next, we define $q+1$ number of games, $Game_k$ where $0 \le k \le q$, and q is the number of key extraction queries made by the adversary. In $Game_k$, the adversary is given semi-functional ciphertext as the challenge ciphertext, and the first k key extraction results are also semi-functional keys, and others are remained like $Game_{Restricted}$. There leaves last game $Game_{Mhiding}$ that is like $Game_q$ except the challenge ciphertext. In $Game_{Mhiding}$ the first component of the challenge ciphertext is a random element of $\mathbb{G}_T$. Then the adversary cannot get any information about the challenge message in

$Game_{Mhiding}$, so that his advantage is information theoretically zero in $Game_{Mhiding}$. The security proof consists of the proofs of indistinguishability between each sequential games.

Theorem 1. *Our HIBE scheme is IND-ID-CPA secure if the group generator $\mathcal{G}$ holds Assumption* 1, 2, 3, 4 *and* 5.

Lemma 1. *If a group generator $\mathcal{G}$ satisfies Assumption* 2, 3 *and* 4, *there is no adversary such that the difference of the advantage in between $Game_{Real}$ and $Game_{Restricted}$ is non-negligible.*

proof. Suppose that $\mathcal{A}$ output identities ID_0 and ID_1 such that $ID_0 \neq ID_1 \ mod \ n$ and $ID_0 = ID_1 \ mod \ p_2$. Then simulator $\mathcal{S}$ can compute a nontrivial factor of n by taking $gcd(ID_0 - ID_1, N)$. Let $gcd(ID_0 - ID_1, N)$=a, and $b = \frac{N}{a}$. Then we consider the following two cases: (Three cases cover all possibilities.)

1. p_1 divides b
2. p_3 divides b
3. p_4 divides b.

In case 1, $\mathcal{S}$ will break Assumption 2. Given instance of Assumption 2, g, X_1X_2, X_3, Y_2Y_3, X_4 and T, $\mathcal{S}$ simulates using g, X_3, and X_4, and then obtains b from $\mathcal{A}$. (Given g, X_3, X_4 $\mathcal{S}$ can simulate with the adversary $\mathcal{A}$.) $\mathcal{S}$ checks $p_1|b$ by testing $g^b = 1$. Next, $\mathcal{S}$ computes $e((X_1X_2)^b, T)$. If $e((X_1X_2)^b, T)$ is the identity of $\mathbb{G}_T$, then $T \in \mathbb{G}_{p_1p_3}$. Otherwise, $T \in \mathbb{G}_{p_1p_2p_3}$.

In case 2, $\mathcal{S}$ will break Assumption 3. Given instance of Assumption 3. X_1, Y_2Y_3, X_3, X_4 and T, $\mathcal{S}$ simulates using X_1, X_3 and X_4, and then obtains b. $\mathcal{S}$ checks $p_3|b$ by testing $X_3^b = 1$. Next, $\mathcal{S}$ checks $e((Y_2Y_3)^b, Y_2Y_3) = e(T^b, Y_2Y_3)$. If the equality holds, then $\mathbb{G}_{p_2}$ part of T is same to Y_2. Otherwise, $\mathbb{G}_{p_2}$ part of T is random.

In case 3, $\mathcal{S}$ will break Assumption 4. Given instance of Assumption 4. X_1, Y_2Y_4, X_3, X_4 and T, $\mathcal{S}$ simulates using X_1, X_3 and X_4, and then obtains b. $\mathcal{S}$ checks $p_4|b$ by testing $X_4^b = 1$. Next, $\mathcal{S}$ checks whether $e((Y_2Y_4)^b, T) = 1_{\mathbb{G}_T}$ or not. If the equality holds, then T is chosen from $\mathbb{G}_{p_4}$. Otherwise, T is chosen from $\mathbb{G}_{p_2p_4}$. □

Lemma 2. *If a group generator $\mathcal{G}$ satisfies Assumption* 1, *there is no adversary such that the difference of the advantage in between $Game_{Restricted}$ and $Game_0$ is non-negligible.*

proof. Simulator $\mathcal{S}$ is given the instance of Assumption 1 ,$\mathbb{G}, n, g, X_3, X_4$ and T.
Setup: $\mathcal{S}$ chooses random integers $b, a_1, \cdots, a_\ell, \alpha \in \mathbb{Z}_n$ and random elements $R_{4,g}$, $R_{4,h}$, $R_{4,u_1}, \cdots$, $R_{4,u_\ell} \in \mathbb{G}_{p_4}$. ($\mathcal{S}$ can compute random elements in $\mathbb{G}_{p_4}$ from randomly exponents of X_4.) It sets and sends $params \leftarrow [\mathbb{G}, n, G = gR_{4,g}, H = g^b R_{4,h}, U_1 = g^{a_1} R_{4,u_1}, \cdots, U_\ell = g^{a_\ell} R_{4,u_\ell}, X_3, X_4, E = e(g, g^\alpha)]$ to $\mathcal{A}$. Keep $[g = g, h = g^b, u_1 = g^{a_1}, \cdots, u_\ell = g^{a_\ell}, w = g^\alpha]$
Query Phase: $\mathcal{S}$ returns to private query for $ID = [I_1, \cdots, I_k]$. Since $\mathcal{S}$ knows MSK, he can generate all private keys.
Challenge: $\mathcal{S}$ is given $ID^* = [I_1^*, \cdots, I_k^*]$ and two messages M_0, M_1 from $\mathcal{A}$. $\mathcal{S}$ tosses a random coin $\beta \in \{0, 1\}$, and returns

$$CT = [C_0 = M_\beta e(T, g^\alpha), C_1 = T^{b + \sum_{i=1}^k a_i I_i^*} R_4', \mathcal{C}_2 = TR_4'']$$

where R_4' and R_4'' are random elements in $\mathbb{G}_{p_4}$. If T is a random element from $\mathbb{G}_{p_1p_4}$, then CT distributes as a normal ciphertext in $Game_{Restricted}$. If T is a random element from $\mathbb{G}_{p_1p_2p_4}$, then CT distributes as a semi-functional ciphertext with $z_c = b + \sum_{i=1}^k a_i I_i^*$ in $Game_0$. Since $z_c \ mod \ p_2$ is not correlated with $b \ mod \ p_1$ and $a_i \ mod \ p_1$, $\mathbb{G}_{p_2}$ part of T^{z_c} is independently random from $params$ and T.
Guess: $\mathcal{S}$ transfers output of $\mathcal{A}$. □

Lemma 3. *If a group generator $\mathcal{G}$ satisfies Assumption 2, 3, there is no adversary such that the difference of the advantage in between $Game_{k-1}$ and $Game_k$ is non-negligible.*

To prove Lemma 3, we use hybrid steps, too. We define a sequence of games $\tilde{Game}_k^{(0)}$, $\tilde{Game}_k^{(1)}$, $\cdots$, $\tilde{Game}_k^{(\ell+1)}$ which locate between $Game_{k-1}$ and $Game_k$. In $\tilde{Game}_k^{(0)}$, $\mathbb{G}_{p_2}$ parts of $Pvk^{(d)}$ are same to $Pvk^{(r)}$ of k-th key query result, and others are remained like $Game_{k-1}$. In $\tilde{Game}_k^{(\tau)}$, $\mathbb{G}_{p_2}$ parts of first τ components of $Pvk^{(d)}$ are independent from first τ components of $Pvk^{(r)}$, and others are remained like $\tilde{Game}_k^{(\tau-1)}$. Then, $\tilde{Game}_k^{(\ell+1)}$ is identically equal to $Game_k$.

Lemma 4. *If a group generator $\mathcal{G}$ satisfies Assumption 2, there is no adversary such that the difference of the advantage in between $Game_{k-1}$ and $\tilde{Game}_k^{(0)}$ is non-negligible.*

proof. Simulator $\mathcal{S}$ is given the instance of Assumption 2 ,$\mathbb{G}, n, g, X_1X_2, X_3, Y_2Y_3, X_4$ and T.
Setup: $\mathcal{S}$ chooses random integers $b, a_1, \cdots, a_\ell, \alpha \in \mathbb{Z}_n$ and random elements $R_{4,g}$, $R_{4,h}$, $R_{4,u_1}, \cdots$, $R_{4,u_\ell} \in \mathbb{G}_{p_4}$. ($\mathcal{S}$ can compute random elements in $\mathbb{G}_{p_4}$ from randomly exponents of X_4.) It sets and sends $params \leftarrow [\mathbb{G}, n, G = gR_{4,g}, H = g^b R_{4,h}, U_1 = g^{a_1} R_{4,u_1}, \cdots, U_\ell = g^{a_\ell} R_{4,u_\ell}, X_3, X_4, E = e(g, g^\alpha)]$ to $\mathcal{A}$. Keep $[g = g, h = g^b, u_1 = g^{a_1}, \cdots, u_\ell = g^{a_\ell}, w = g^\alpha]$
Query Phase: Since $\mathcal{S}$ knows MSK, he can generate all normal private keys. For first i-th ($i < k$) queries, $\mathcal{S}$ generates normal private keys, and multiplies a random power of Y_2Y_3 to every component of keys, and then he returns to the adversary. These keys are distributed as semi-functional keys. For $i > k$ case, $\mathcal{S}$ returns normal keys. For k-th query for $ID = [I_1, \cdots, I_j]$, $\mathcal{S}$ chooses a random integer $t \in \mathbb{Z}_n$ and random elements $R_3^{(d)}, R_{3,j+1}^{(d)}, \cdots, R_{3,\ell}^{(d)}$, $R_3^{(r)}$, $R_{3,j+1}^{(r)}$, $\cdots$, $R_{3,\ell}^{(r)} \in \mathbb{G}_{p_3}$ and respectively sets $Pvk_{ID}^{(d)}$ and $Pvk_{ID}^{(r)}$ as follows:

$$Pvk_{ID}^{(d)} = \begin{cases} K_1^{(d)} \leftarrow & T, \\ K_2^{(d)} \leftarrow & wT^{b+\sum_{i=1}^{j} a_i I_i} R_3^{(d)}, \\ E_i^{(d)} \leftarrow & T^{a_i} R_{3,i}^{(d)}], \quad \forall\ i \in [j+1, \ell] \end{cases}$$

$$Pvk_{ID}^{(r)} = \begin{cases} K_1^{(r)} \leftarrow & T^t, \\ K_2^{(r)} \leftarrow & T^{t(b+\sum_{i=1}^{j} a_i I_i)} R_3^{(r)}, \\ E_i^{(r)} \leftarrow & T^{ta_i} R_{3,i}^{(r)} \quad \forall\ i \in [j+1, \ell]. \end{cases}$$

If $T \in \mathbb{G}_{p_1p_3}$ above is a normal key in $Game_{k-1}$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then each $\mathbb{G}_{p_2}$ part of $Pvk^{(d)}$ is independently random from $params$ and T since $b + \sum_{i=1}^{j} a_i I_i \mod p_2$ and $a_i \mod p_2$ for $i \in [j+1, \ell]$ are independently random from $params$ and T. $\mathbb{G}_{p_2}$ part of $Pvk^{(r)}$ is same to $Pvk^{(d)}$, so that this is a key in $\tilde{Game}_k^{(0)}$. Note that $z_k = b + \sum_{i=1}^{j} a_i I_i$.
Challenge: $\mathcal{S}$ is given $ID^* = [I_1^*, \cdots, I_k^*]$ and two messages M_0, M_1 from $\mathcal{A}$. $\mathcal{S}$ tosses a random coin $\beta \in \{0, 1\}$, and returns the challenge ciphertext

$$[M_\beta e(X_1X_2, w),\ (X_1X_2)^{b+\sum_{i=1}^{k} a_i I_i^*} R_4',\ (X_1X_2)R_4'']$$

where R_4' and R_4'' are chosen at random from $\mathbb{G}_{p_4}$. Note that $z_c = b + \sum_{i=1}^{k} a_i I_i^*$. Since for all ID queried by $\mathcal{A}$, $ID \mod p_2$ is not equal to $ID^* \mod p_2$, $z_c \mod p_2$ is independent random from $z_k \mod p_2$, and $a_i \mod p_2$ for $i \in [j+1, \ell]$ used in the k-th key query. Hence, all randomness used in the challenge ciphertexts are independently random from all other

randomness used in the game. If $\mathcal{S}$ generates the corresponding semi-functional ciphertext of k-th key query, and tests whether k-th key is semi-functional key, then decryption will always work without respect to that the k-th key is semi-functional key or not since $z_c = z_k$.
Guess: $\mathcal{S}$ transfers output of $\mathcal{A}$. □

Lemma 5. *If a group generator $\mathcal{G}$ satisfies Assumption 3, there is no adversary such that the difference of the advantage in between $\tilde{Game}_k^{(0)}$ and $\tilde{Game}_k^{(1)}$ is non-negligible.*

proof. Simulator $\mathcal{S}$ is given the instance of Assumption 3 ,$\mathbb{G}, n, g, X_1, Y_2Y_3, X_3$, X_4 and T.
Setup: $\mathcal{S}$ chooses random integers $b, a_1, \cdots, a_\ell, \alpha$ and random elements $R_{4,g}$, $R_{4,h}$, $R_{4,u_1}, \cdots$, $R_{4,u_\ell} \in \mathbb{G}_{p_4}$. ($\mathcal{S}$ can compute random elements in $\mathbb{G}_{p_4}$ from randomly exponents of X_4.) It sets and sends $params \leftarrow [\mathbb{G}, n, G = gR_{4,g}, H = g^b R_{4,h}, U_1 = g^{a_1} R_{4,u_1}, \cdots, U_\ell = g^{a_\ell} R_{4,u_\ell}$, X_3, X_4, $E = e(g, g^\alpha)]$ to $\mathcal{A}$. Keep $[g = g, h = g^b, u_1 = g^{a_1}, \cdots, u_\ell = g^{a_\ell}, w = g^\alpha]$
Query Phase: Since $\mathcal{S}$ knows MSK, he can generate all normal private keys. For first i-th ($i < k$) queries, $\mathcal{S}$ generates normal private keys, and multiplies a random power of Y_2Y_3 to every component of keys, and then he returns to the adversary. These keys are distributed as semi-functional keys. For $i > k$ case, $\mathcal{S}$ returns normal keys. For k-th query for $ID = [I_1, \cdots, I_j]$, $\mathcal{S}$ chooses random integers $r_1, r_2, t, t_{j+1}, \cdots, t_\ell \in \mathbb{Z}_n$ and random elements $R_3^{(d)}, R_{3,j+1}^{(d)}, \cdots, R_{3,\ell}^{(d)}, R_3^{(r)}, R_{3,j+1}^{(r)}, \cdots, R_{3,\ell}^{(r)} \in \mathbb{G}_{p_3}$ and respectively sets $Pvk_{ID}^{(d)}$ and $Pvk_{ID}^{(r)}$ as follows:

$$Pvk_{ID}^{(d)} = \begin{cases} K_1^{(d)} \leftarrow & g^{r_1}(Y_2Y_3), \\ K_2^{(d)} \leftarrow & w(h\prod_{i=1}^{j} u_i^{a_i})^{r_1}(Y_2Y_3)^t R_3^{(d)}, \\ E_i^{(d)} \leftarrow & u_i^{r_1 a_i}(Y_2Y_3)^{t_i} R_{3,i}^{(d)}, \quad \forall\ i \in [j+1, \ell] \end{cases}$$

$$Pvk_{ID}^{(r)} = \begin{cases} K_1^{(r)} \leftarrow & g^{r_2}T, \\ K_2^{(r)} \leftarrow & (h\prod_{i=1}^{j} u_i^{a_i})^{r_2}(Y_2Y_3)^t R_3^{(d)}, \\ E_i^{(r)} \leftarrow & u_i^{r_2 a_i}(Y_2Y_3)^{t_i} R_{3,i}^{(r)}. \quad \forall\ i \in [j+1, \ell] \end{cases}$$

If $T = Y_2Y_3'$, $\mathbb{G}_{p_2}$ part of each component of $Pvk^{(r)}$ is same to $\mathbb{G}_{p_2}$ part of the corresponding component of $Pvk^{(d)}$, so that this is a key in $\tilde{Game}_k^{(0)}$. If T is random element in $\mathbb{G}_{p_2p_3}$, then above is a key in $\tilde{Game}_k^{(1)}$.
Challenge: $\mathcal{S}$ is given $ID^* = [I_1^*, \cdots, I_k^*]$ and two messages M_0, M_1 from $\mathcal{A}$. $\mathcal{S}$ tosses a random coin $\beta \in \{0, 1\}$, and returns the challenge ciphertext

$$[M_\beta e(g, g^\alpha)^s,\ (h\prod_{i=1}^{j} u_i^{a_i})^s (Y_2Y_3)^{s'} R_4',\ g^s (Y_2Y_3)^{s''} R_4'']$$

where R_4' and R_4'' are random elements in $\mathbb{G}_{p_4}$ and $s, s', s'' \in \mathbb{Z}_n$ are random integers.
Guess: $\mathcal{S}$ transfers output of $\mathcal{A}$. □

Similarly we can prove indistinguishability between $\tilde{Game}_k(\tau)$ and $\tilde{Game}_k(\tau+1)$ for $\tau \in [1, \ell]$. Simulator can generate all normal keys, semi-functional keys, the challenge ciphertext, and $Pvk_{ID}^{(d)}$ for k-th query for ID using the instance of Assumptions 3 as in Lemma 5. Then, $\mathcal{S}$ computes the $(\tau+1)$-th component of $Pvk_{ID}^{(r)}$ using T for its $\mathbb{G}_{p_2p_3}$ part. Since there is no technical difference from the security proof of Lemma 5, we give following lemma without proof.

Lemma 6. *If a group generator $\mathcal{G}$ satisfies Assumption 3, there is no adversary such that the difference of the advantage in between $\tilde{Game}_k^{(1)}$ and $\tilde{Game}_k^{(\ell+1)} = Game_k$ is non-negligible.*

Lemma 4, 5 and 6 imply Lemma 3.

Lemma 7. *If a group generator $\mathcal{G}$ satisfies Assumption 5, there is no adversary such that the difference of the advantage in between $Game_q$ and $Game_{Mhiding}$ is non-negligible.*

proof. Simulator $\mathcal{S}$ is given the instance of Assumption 5 ,$\mathbb{G}, n, g, X_1X_2, X_3,\ Y_1Y_2, Z_2Z_3, X_4$ and T.
Setup: $\mathcal{S}$ chooses random integers $b, a_1, \cdots, a_\ell, \alpha \in \mathbb{Z}_n$ and random elements $R_{4,g}$, $R_{4,h}$, $R_{4,u_1}, \cdots$, $R_{4,u_\ell} \in \mathbb{G}_{p_4}$. ($\mathcal{S}$ can compute random elements in $\mathbb{G}_{p_4}$ from randomly exponents of X_4.) It sets and sends $params \leftarrow [\mathbb{G}, ns, G = gR_{4,g}, H = g^bR_{4,h}, U_1 = g^{a_1}R_{4,u_1}, \cdots, U_\ell = g^{a_\ell}R_{4,u_\ell}, X_3, X_4, E = e(g, X_1X_2)]$ to $\mathcal{A}$. Keep $[g = g, h = g^b, u_1 = g^{a_1}, \cdots, u_\ell = g^{a_\ell}]$. Then an unknown master secret key w is X_1.
Query Phase: $\mathcal{S}$ returns to private query for $ID = [I_1, \cdots, I_j]$. $\mathcal{S}$ chooses random integers r_1, t, t_{j+1}, $\cdots$, t_ℓ, r_2, s, s_{j+1}, $\cdots$, $s_\ell \in \mathbb{Z}_n$ and random elements $R_3^{(d)}$, $R_3^{\prime(d)}$, $R_{3,j+1}^{(d)}$, $\cdots$,$R_{3,\ell}^{(d)}$, $R_3^{(r)}$, $R_3^{\prime(r)}$, $R_{3,j+1}^{(r)}$, $\cdots$, $R_{3,\ell}^{(r)} \in \mathbb{G}_{p_3}$. X_3 can be used for generating random elements in $\mathbb{G}_{p_3}$. He sets a semi-functional key $Pvk_{ID}^{(d)}$ and $Pvk_{ID}^{(r)}$ as follows:

$$Pvk_{ID}^{(d)} = \begin{cases} K_1^{(d)} \leftarrow & g^{r_1}(Z_2Z_3)^t R_3^{(d)}, \\ K_2^{(d)} \leftarrow & X_1X_2(h\prod_{i=1}^{j} u_i^{I_i})^{r_1} R_3^{\prime(d)}, \\ E_i^{(d)} \leftarrow & u_i^{r_1}(Z_2Z_3)^{t_i} R_{3,i}^{(d)}, \quad \forall\ i \in [j+1, \ell] \end{cases}$$

$$Pvk_{ID}^{(r)} = \begin{cases} K_1^{(r)} \leftarrow & g^{r_2}(Z_2Z_3)^s R_3^{(r)}, \\ K_2^{(r)} \leftarrow & (h\prod_{i=1}^{j} u_i^{I_i})^{r_2}(Z_2Z_3) R_3^{\prime(r)}, \\ E_i^{(r)} \leftarrow & u_i^{r_2}(Z_2Z_3)^{s_i} R_{3,i}^{(d)}, \quad \forall\ i \in [j+1, \ell] \end{cases}$$

$\mathcal{S}$ sends these to $\mathcal{A}$.
Challenge: $\mathcal{S}$ is given $ID^* = I_1^*, \cdots, I_k^*$ and two messages M_0, M_1 from $\mathcal{A}$. $\mathcal{S}$ tosses a random coin $\beta \in \{0, 1\}$, and returns the challenge ciphertext

$$[M_\beta T,\ (Y_1Y_2)^{b+\sum_{i=1}^{k} a_i I_i^*} R_4',\ Y_1Y_2R_4'']$$

where R_4' and R_4'' are random elements in $\mathbb{G}_{p_4}$. Since $b + \sum_{i=1}^{k} a_i I_i^*$ *mod* p_2 is independent from $a_i \mod p_2$ for $i \in [1, \ell]$ and $b \mod p_2$, $\mathbb{G}_{p_2}$ parts of C_1 and C_2 are independent random elements from *params*. If T is a random element from $\mathbb{G}_T$, then CT distributes as a ciphertext in $Game_{Mhiding}$. If $T = e(X_1, Y_1)$, then CT distributes as a semi-functional ciphertext with $z_c = b + \sum_{i=1}^{k} a_i I_i^*$ in $Game_q$.
Guess: $\mathcal{S}$ transfers output of $\mathcal{A}$. □

Proof of Theorem 1
In $Game_{Mhiding}$ the adversary cannot get information about the challenge messages since the challenge message is multiplied by a random element in the challenge ciphertext. Therefore the advantage of the adversary in $Game_{Mhiding}$ is information theoretically zero. By Lemma 1, 2, 3 and Lemma 7, theorem is completed. □

We also uses a hybrid steps for proving the anonymity. Similarly to the proof of the confidentiality we define a sequence of games $Game_{Real}$, $Game_{Restricted}$, $Game_1$, $\cdots$, $Game_q$, $Game_{Mhiding}$. Additionally we define $Game_{Random}$ that C_1 and C_2 in the challenge ciphertexts are independent random elements in $\mathbb{G}_{p_1p_2p_4}$, others are remained like $Game_{Mhiding}$. The

adversary in $Game_{Random}$ cannot get any information about the identity from the challenge ciphertext, so that his advantage is information theoretically zero. We can show that these games are indistinguishable step by step.

Theorem 2. *Our HIBE scheme is ANON-ID-CPA secure if a group generator $\mathcal{G}$ holds Assumption* 1*,* 2*,* 3*,* 4*,* 5 *and* 6*.*

Lemma 8. *If a group generator $\mathcal{G}$ satisfies Assumption* 1*,* 2*,* 3*,* 4 *and* 5*, there is no adversary such that the difference of the advantage in between $Game_{Real}$ and $Game_{Mhiding}$ is non-negligible.*

The proof of Lemma 8 is basically same to the proof of Theorem 1. We defined $Game_{Real}$ and $Game_{Mhiding}$ as ANON-ID-CPA games. The differences among a sequence of games as ANON-ID-CPA is essentially same to the differences among a sequence of games as ANON-ID-CPA. Therefore the proof of Theorem 1 essentially implies Lemma 8.

Lemma 9. *If a group generator $\mathcal{G}$ satisfies Assumption* 6*, there is no adversary such that the difference of the advantage in between $Game_{Mhiding}$ and $Game_{Random}$ is non-negligible.*

proof. Suppose that there exists an adversary $\mathcal{A}$ such that the difference of the advantage in between $Game_{Mhiding}$ and $Game_{Random}$ is non-negligible. Now we describe that the simulator $\mathcal{S}$ breaks Assumption 6 by using $\mathcal{A}$ with non-negligible advantage. $\mathcal{S}$ receives the instance of Assumption 6, X_1X_4, Y_1Y_2, Z_2, Z_3, Z_4, $W_1W_2W_4$, and T.
Setup. It chooses random integers $a_1, \cdots, a_\ell, b \in \mathbb{Z}_n$ and random elements $Z_{4,h}, Z_{4,u_i} \in G_{p_4}$. It sets $params = [\mathbb{G}, n, G = X_1X_4,\ H = (X_1X_4)^b Z_{4,h},\ U_1 = (X_1X_4)^{a_1} Z_{4,u_1},\ \cdots,\ U_\ell = (X_1X_4)^{a_\ell} Z_{4,u_\ell},\ Z_3,\ Z_4,\ E = e(X_1X_4, Y_1Y_2)]$ and sends it to $\mathcal{A}$. Then unknown secret elements are $[g = X_1, h = X_1^b, u_1 = X_1^{a_1}, \cdots, u_\ell = X_1^{a_\ell}, w = Y_1]$.
Query Phase1. When $\mathcal{A}$ queries the private key for $ID = [I_1, \cdots, I_k]$, $\mathcal{S}$ chooses random integers $r_1, r_2 \in \mathbb{Z}_N$ and random elements $Z_2^{\prime(d)}$, $Z_2^{\prime\prime(d)}$, $Z_{2,k+1}^{(d)}$, $\cdots$, $Z_{2,\ell}^{(d)}$, $Z_2^{\prime(r)}$, $Z_2^{\prime\prime(r)}$, $Z_{2,k+1}^{(r)}$, $\cdots$, $Z_{2,\ell}^{(r)} \in \mathbb{G}_{p_2}$, $Z_3^{\prime(d)}$, $Z_3^{\prime\prime(d)}$, $Z_{3,k+1}^{(d)}$, $\cdots$, $Z_{3,\ell}^{(d)}$, $Z_3^{\prime(r)}$, $Z_3^{\prime\prime(r)}$, $Z_{3,k+1}^{(r)}$, $\cdots$, $Z_{3,\ell}^{(r)} \in \mathbb{G}_{p_3}$ to generate semi-functional key for $ID|_k$. It sets $Pvk_{ID}^{(d)}$ and $Pvk_{ID}^{(r)}$ as follows:

$$
\begin{aligned}
K_1^{(d)} &\leftarrow (Y_1Y_2)^{r_1} Z_2^{\prime(d)} Z_3^{\prime(d)},\\
K_2^{(d)} &\leftarrow (Y_1Y_2)((Y_1Y_2)^b \textstyle\prod_{i=1}^k (Y_1Y_2)^{a_i I_i})^{r_1} Z_2^{\prime\prime(d)} Z_3^{\prime\prime(d)},\\
E_i^{(d)} &\leftarrow (Y_1Y_2)^{a_i r_1} Z_{2,i}^{(d)} Z_{3,i}^{(d)} \quad \forall i \in [k+1, \ell].\\
K_1^{(r)} &\leftarrow (Y_1Y_2)^{r_2} Z_2^{\prime(r)} Z_3^{\prime(r)},\\
K_2^{(r)} &\leftarrow ((Y_1Y_2)^b \textstyle\prod_{i=1}^k (Y_1Y_2)^{a_i I_i})^{r_2} Z_2^{\prime\prime(r)} Z_3^{\prime\prime(r)},\\
E_i^{(r)} &\leftarrow (Y_1Y_2)^{a_i r_2} Z_{2,i}^{(r)} Z_{3,i}^{(r)} \quad \forall i \in [k+1, \ell]
\end{aligned}
$$

Then the above keys are well-formed with randomness as if $\bar{r_1} = r_1 \log_{X_1} Y_1$ for $Pvk_{ID}^{(d)}$, $\bar{r_2} = r_2 \log_{X_1} Y_1$ for $Pvk_{ID}^{(r)}$.
Challenge. $\mathcal{A}$ gives a message M and two identities $ID^{(0)} = [I_1^{(0)}, \cdots, I_{k_0}^{(0)}]$ and $ID^{(1)} = [I_1^{(1)}, \cdots, I_{k_1}^{(1)}]$ to $\mathcal{S}$. $\mathcal{S}$ chooses a random coin $\beta \in \{0, 1\}$, and random elements $Z_2', Z_2'' \in \mathbb{G}_{p_2}$, $Z_3' \in \mathbb{G}_{p_3}$, $Z_4'' \in \mathbb{G}_{p_4}$, $R \in \mathbb{G}_T$, and then it sets the challenge ciphertext as follows:

$$
[R,\ (W_1W_2W_4)^{b + \sum_{i=0}^{k_\beta} a_i I_i^{(\beta)}} Z_2'' Z_4'',\ T].
$$

Table 1. Anonymous HIBE schemes

	size in *params*	size in Pvk	size in CT	Security Model	# of primes in group order
[9]	$O(\ell^2)$	$O(\ell^2)$	$O(\ell)$	Selective	1
[22]	$O(\ell)$	$O((\ell-k)k)$	$O(\ell)$	Selective	2
[19]	$O(\ell)$	$O(\ell-k)$	4	Selective	2
ours	$O(\ell)$	$O(\ell-k)$	3	Full	4

ℓ: the maximum depth of hierarchy,
k: a depth of a corresponding identity,

A random element in $\mathbb{G}_T$ can be generated by raising $e(X_1X_4Z'_2Z'_3, X_1X_4Z'_2Z'_3)$ to a random integer in $\mathbb{Z}_n$.

If $T = W_1W'_2W'_4$ then CT distributes as in $Game_{Mhiding}$. Otherwise, since $W_1W_2W_4$ is chosen independent at random, CT distributes as in $Game_{Random}$.

Query Phase 2. $\mathcal{A}$ adaptively issues key extraction queries and $\mathcal{S}$ replies as Query Phase1.

Guess. $\mathcal{A}$ outputs a bit β', then $\mathcal{S}$ also return the same bit β' as its guess. □

Proof of Theorem 2

In $Game_{Random}$ the adversary cannot get information about the challenge ID since the challenge ciphertext distributes as random. Therefore the advantage of the adversary in $Game_{Random}$ is information theoretically zero. By Lemma 8 and Lemma 9, theorem is completed. □

6 Related Works-Anonymous HIBE

The concepts of anonymous HIBE scheme were introduced by Abdalla et. al. [1]. A first realization of anonymous HIBE scheme was proposed by Boyen and Waters [9]. They attained anonymity under the decisional linear assumption. Shi and Waters proposed a delegatable hidden-vector encryption (dHVE) whose definition is a generalization of anonymous HIBE [22]. First anonymous HIBE scheme with constant size ciphertexts was proposed by Seo et. al.[19]. They embedded Boneh, Boyen, and Goh's HIBE scheme with short ciphertexts to the subgroup $\mathbb{G}_p$ of the composite order bilinear group $\mathbb{G}_p \times \mathbb{G}_q$, and blinds ID information using random elements of subgroup $\mathbb{G}_q$. Recently Ducas proposed new constructions for anonymous HIBE using asymmetric pairing [14]. All prior anonymous HIBE schemes, however, were proved only in the weaker *selective* security notion, that is, the adversary should select the target ID before she see the system parameters. We give comparisons with our construction in the table 1. Recently, Caro et al. [12] proposed a fully secure anonymous HIBE scheme with constant ciphertexts, which attains the same performance as our HIBE scheme. We should note that this paper and [12] are definitely independent results.

7 Conclusion

In this paper we proposed a fully secure anonymous hierarchical ID-based encryption scheme with constant size ciphertexts in composite order bilinear group of four primes, and proved the security under six static assumptions. Our construction satisfies full security, anonymity, and constant size ciphertexts, together, so that it is able to be used as a primitive in public-key searchable encryption fields efficiently. We leave efficient constructions in prime order

group under simple assumptions, such as the bilinear Diffie-Hellman assumption and linear assumption as an interesting open problem.

References

1. M. Abdalla, M. bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 205–222. Springer-Verlag, 2005.
2. D. Boneh and X. Boyen. Efficient selective-id identity based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, 2004.
3. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *SIAM J. Comput.*, volume 36, pages 1301–1328, Philadelphia, PA, USA, December 2006. Society for Industrial and Applied Mathematics.
4. D. Boneh, G. D. Crecenzo, R. Ostrovsky, and G. Persiano. Public-key encryption with keyword search. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522. Springer-Verlag, 2004.
5. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 19–23. Springer-Verlag, 2001.
6. D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC 2005*, volume 3378 of *LNCS*. Springer-Verlag, 2005.
7. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC 2007*, volume 4392 of *LNCS*, pages 535–554. Springer-Verlag, 2007.
8. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security-CCS 2005*. ACM Press, 2005.
9. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO 2006*, volume 4117 of *LNCS*, pages 290–307. Springer-Verlag, 2006.
10. J. Camenisch, M. Kohlweiss, and C. Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *PKC 2009*, volume 5443 of *LNCS*, pages 481–500. Springer, 2009.
11. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT 2003*, volume 2656 of *LNCS*. Springer-Verlag, 2003.
12. A. D. Caro, V. Iovino, and G. Persiano. Fully secure anonymous hibe and scert-key anonymous ibe with short ciphertext. In *Pairing*, volume 6487 of *LNCS*, pages 347–366. Springer, 2010.
13. D. Davis, F. Monrose, and M. K. Reiter. Time-scoped searching of encrypted audit logs. In *ICICS 2004*, pages 532–545, 2004.
14. L. Ducas. Anonymity from asymmetry: New constructions for anonymous hibe. In *CT-RSA 2010*, volume 5985 of *LNCS*, pages 148–164. Springer, 2010.
15. C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 149–155. Springer-Verlag, 2002.
16. J. Horwitz and B. Lynn. Towards hierarchical identity-based encryption. In *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 466–481. Springer-Verlag, 2002.
17. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, LNCS. Springer-Verlag, 2008.
18. A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, 2010.
19. J. H. Seo, T. Kobayashi, M. Ohkubo, and K. Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In G. Tsudik and S. Jarecki, editors, *PKC 2009*, volume 5443 of *LNCS*, pages 215–234. Springer-Verlag, 2009.
20. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, LNCS, pages 47–53. Springer, 1984.
21. E. Shi, J. Bethencourt, H. T.-H. Chan, D. X. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy*, pages 350–364. IEEE Computer Society, 2007.
22. E. Shi and B. Waters. Delegating capabilities in predicate encryption systems. In *ICALP 2008*, volume 5126 of *LNCS*, pages 560–578. Springer-Verlag, 2008.
23. B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer-Verlag, 2009.

24. B. Waters, D. Balfanz, G. Durfee, and D. Smetters. Building an encrypted and searchanle audit log. In *NDSS 2004*, 2004.
25. D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya. Id-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In *ACM Conference on Computer and Communications Security-CCS 2004*, pages 356–363, 2004.