# Compact CCA2-secure Hierarchical Identity-Based Broadcast Encryption for Fuzzy-entity Data Sharing 

Weiran Liu ${ }^{1}$, Jianwei Liu ${ }^{1}$, Qianhong Wu ${ }^{1}$, Bo Qin ${ }^{2}$, David Naccache ${ }^{3}$, and Houda Ferradi ${ }^{4}$<br>${ }^{1}$ School of Electronic and Information Engineering, Beihang University, XueYuan Road No.37, Haidian District, Beijing, China<br>liuweiran900217@gmail.com, liujianwei@buaa.edu.cn, qianhong.wu@buaa.edu.cn<br>${ }^{2}$ School of Information, Renmin University of China, ZhongGuanCun Street No. 59, Haidian District, Beijing, China<br>bo.qin@ruc.edu.cn<br>${ }^{3}$ Département d'Informatique 45 rue dúlm, École normale supérieure, Paris cedex 05, f-75230, France david.naccache@ens.fr, Houda.Ferradi@ens.fr


#### Abstract

With the advances of cloud computing, data sharing becomes easier for large-scale enterprises. When deploying privacy and security schemes in data sharing systems, fuzzy-entity data sharing, entity management, and efficiency must take into account, especially when the system is asked to share data with a large number of users in a tree-like structure. (Hierarchical) Identity-Based Encryption is a promising candidate to ensure fuzzy-entity data sharing functionalities while meeting the security requirement, but encounters efficiency difficulty in multiuser settings. This paper proposes a new primitive called Hierarchical Identity-Based Broadcast Encryption (HIBBE) to support multi-user data sharing mechanism. Similar to HIBE, HIBBE organizes users in a tree-like structure and users can delegate their decryption capability to their subordinates. Unlike HIBE merely allowing a single decryption path, HIBBE enables encryption to any subset of the users and only the intended users (and their supervisors) can decrypt. We define Ciphertext Indistinguishability against Adaptively Chosen-Identity-Vector-Set and Chosen-Ciphertext Attack (IND-CIVS-CCA2) for HIBBE, which capture the most powerful attacks in the real world. We achieve this goal in the standard model in two steps. We first construct an efficient HIBBE Scheme (HIBBES) against Adaptively Chosen-Identity-Vector-Set and Chosen-Plaintext Attack (IND-CIVS-CPA) in which the attacker is not allowed to query the decryption oracle. Then we convert it into an IND-CIVS-CCA2 scheme at only a marginal cost, i.e., merely adding one on-the-fly dummy user at the first depth of hierarchy in the basic scheme without requiring any other cryptographic primitives. Our CCA2-secure scheme natively allows public ciphertext validity test, which is a useful property when a CCA2-secure HIBBES is used to design advanced protocols and auditing mechanisms for HIBBE-based data sharing.


Keywords: Hierarchical Identity-Based Broadcast Encryption; Adaptive Security; Chosen-ciphertext Security; Fuzzy-entity Data Sharing

## 1 Introduction

The rapid development of "Cloud Computing" have brought great convenience for on-demand data sharing. Nowadays, large-scale enterprises choose to acquire cloud storage services from a cloud service provider, or establishing its own cloud data center for cost-effective data sharing. In this paradigm, individual staff in such an enterprise can easily acquire useful data, while sharing data to its superiors, colleagues, and subordinates in an on-demand manner. This significantly improves the communication efficiency, lower the data sharing expenses, thus brings benefits to the enterprises.

Due to its openness, data sharing system is always deployed in a hostile environment and vulnerable to a number of security threats [25]. Among all, data privacy, legal access, and data authenticity are the main security concerns in data sharing systems [12]. The above security issue can be respectively addressed with the help of traditional cryptographic tools, e.g., encryption, message authentication code (MAC), digital signatures. However, leveraging these cryptographic tools into large-scale data sharing systems may bring additional difficulties when taking into account other issues, such as fuzzyentity data sharing, effective entity management, and efficiency.

Traditional cryptographic tools allow data encryption and data authentication after explicitly knowing the receivers' public yet random information, i.e., public keys. When the personnel structure of the company changes, which rapidly happens in a large-scale enterprise, fuzzy-entity data sharing is needed so that stuffs' can share data without knowing the receivers' public keys, but the recognizable identities. Identity-Based Encryption (IBE), introduced by Shamir [37], allows one to securely communicate with others if he/she knows their public identities. In IBE, users' recognizable identities such as their social security numbers, IPs or email addresses, are used as their public keys. A Private Key Generator (PKG) is used to generate secret keys associated with the users' public identities. One can encrypt to any user by specifying its recognizable identity and only the intended user can decrypt.

While IBE supports fuzzy-entity data sharing in the enterprise, it faces the difficulty of inefficient entity management. In IBE systems, every entity should ask PKG for obtaining a secret key associated with its own identities. However, the number of users in a data sharing system could be huge [25]. With the number of users in the system increase, PKG may be busy with generating secret keys for replying secret key obtaining requests from the users. A method of sharing PKG's burden is required. Hierarchical IBE (HIBE) extends IBE to endow a large number of users with a delegation mechanism. HIBE [20] organizes users in a tree-like structure which is consistent with the structure of large-scale enterprises and organizations [16, 42]. PKG's burden is shared by upper-level users who can delegate secret keys to their subordinates. In the encryption process, the sender associates the ciphertext with an identity vector instead of a single identity. Then only the users whose identities appear in the specified identity vector can decrypt.

When applying HIBE in an enterprise or an organization for data sharing, one should also consider efficiency aspects, that is, the computation and communication costs in different data sharing situations. In such application scenario, individual stuff may have to simultaneously communicate and share data with multiple users in hierarchical organizations. For example, the enterprise may cooperate with a number of professors from different laboratories in a university to develop a new software system. The enterprise can separately encrypt to these professors by specifying their respective decryption paths. However, this trivial solution incurs heavy encryption burden and long ciphertexts. Another example comes from the cloud-based electronic health record system, where medical stuff should share patients' electronic health record with chief/assistant doctors in distinct departments [33]. Applying existing HIBE schemes in such systems is a reasonable solution. However, HIBE gradually becomes inefficient when the number of involved departments increases. We are interested in more practical solutions to such applications.

### 1.1 Our Contributions

We propose a new cryptographic primitive called Hierarchical Identity-Based Broadcast Encryption (HIBBE). Users in a tree-like structure can delegate their decryption capabilities to their subordinates, so that the burden of the PKG can be shared when the system hosts a large number of users. One can encrypt to any subset of the users and only the intended ones and their supervisors can decrypt.

We define the security notion for HIBBE, named Ciphertext Indistinguishability against Adaptively Chosen-Identity-Vector-Set and Chosen-Ciphertext attack (IND-CIVS-CCA2). In this notion, the attacker is simultaneously allowed to adaptively query for the secret keys of users recognized by identity vectors of its choice and to issue decryption queries for receiver identity vector sets at wish. Even such an attacker cannot distinguish the encrypted messages, provided that the attacker does not query for the secret keys of the target users or their supervisors. Clearly, this definition captures the most powerful attacks on HIBBE in the real world.

We obtain an IND-CIVS-CCA2 scheme in the standard model (without using random oracles) in two steps. We first construct an HIBBE Scheme (HIBBES) against Adaptively Chosen-Identity-VectorSet and Chosen-Plaintext Attack (IND-CIVS-CPA) in the standard model, in which the attacker is not allowed to issue decryption queries. Then, at merely marginal cost, we convert the basic scheme into an IND-CIVS-CCA2 scheme by adding only one on-the-fly dummy user, rather than adding one hierarchy of users in existing conversions from a CPA-secure hierarchical encryption scheme to a CCA2secure one. Both schemes have constant size ciphertext and are efficient in terms of communications and data sharing in multi-receiver situations. This novel cryptographic scheme suitably meets the security and efficiency requirement of large-scale enterprises, including fuzzy-entity data sharing, entity management, and efficiency.

Compared with the preliminary version [31] of the paper, in this extended work we give the formal security proof of the CPA security of the basic scheme; we further convert the CPA-secure HIBBES into a CCA2-secure HIBBES with compact design in the sense that the conversion does not require any other cryptographic primitives; we formally prove that the resulting scheme is CCA2-secure in the standard model. Our CCA2-secure HIBBES allows public ciphertext validity test which is useful for a third party, e.g., a firewall, to filter invalid spams and for system designers to design advanced protocols from HIBBE, e.g., publicly verifiable HIBBE allowing auditing for cloud data center [13, 38], and data authentication of HIBBE-encrypted digital contents [26].

### 1.2 Related Work

Identity-Based Encryption. Since the concept of Identity-Based Encryption (IBE) was introduced by Shamir [37], it took a long time for researchers to construct a practical and fully functional IBE Scheme (IBES). In 2001, Boneh and Franklin [3, 4] precisely defined the security model of IBE and proposed the first practical IBES by using bilinear pairings. In the Boneh-Franklin security model, the adversary can adaptively request secret keys for the identities of its choice and can choose the challenge identity it wants to attack at any point during the key-requesting process, provided that the secret key for the challenging identity is not queried. The security of their IBES $[3,4]$ requires cryptographic hash functions to be modeled as random oracles. Canetti et al. [10, 11] formalized a slightly weaker security notion, called selective-ID security, in which the adversary must disclose the challenge identity before the public parameters are generated. They exhibited a selective-ID secure IBES without using random oracles. Since then, more practical IBES have been proposed that are shown to be secure without random oracles in the selective-ID security model [1] or in the standard security model [39]. These schemes are secure against CPA. Interestingly, some recent works [8, 9, 11] showed CPA-secure IBES can be used to construct regular Public-Key Encryption systems with CCA2 security. Canetti, Halevi and Katz [11] exhibited a generic conversion by adding a one-time signature scheme and hash the signature parameters as a special "identity" in encryption. Boneh and Katz [8] later presented a more efficient construction using a MAC to replace the one-time signature. More recently, Boyen et al. [9] introduced a new technique that can directly obtain CCA2 security from some particular IBES without extra cryptographic primitives. Park et al. [34] proposed a concrete CCA2-secure IBES with a tight security reduction in the random oracle model.

Broadcast Encryption. In Broadcast Encryption (BE) [18], a dealer is employed to generate and distribute decryption keys for users. A sender can encrypt to a subset of the users and only the privileged users can decrypt. This functionality models flexible secure one-to-many communication scenarios [35]. Since the BE concept was introduced in 1994 [18], many BE Schemes have been proposed to gain more preferable properties. We mention just a few of those properties, such as "Stateless Receivers" (after getting the broadcast secret keys, users do not need to update them) [17, 22], "Fully Collusion Resistant" (even if all users except the receiver set collude, they can obtain no information about the plaintext) [5], "Dynamic" (the dealer can dynamically recruit new members while the other members will not be affected) [15], "Anonymity" (a receiver does not need to know who the other receivers are when decrypting ciphertexts) [30], and "Contributory Broadcast" (Anyone can send messages to any subset of the group members without a trusted key server) [41].

Identity-Based Broadcast Encryption. Identity-Based Broadcast Encryption (IBBE) incorporates the idea of BE into IBE and recognizes the users in a BES with their identities, instead of indexes assigned by the system. When one needs to send confidential messages to multiple users, the sender in IBBE can efficiently encrypt the message once to multiple users and simply broadcasts the resulting ciphertext. Fully functional IBBE was formalized and realized by Delerablée with constant size ciphertexts and secret keys [14], although it is only selective-ID secure in the random oracle model. The up-to-date IBBE Schemes $[21,36,27]$ are shown to be secure in the standard security model.

Hierarchical Identity-Based Encryption. Horwitz and Lynn [23] first proposed the concept of HIBE and presented a two-level HIBES in the same article. The first fully functional HIBE construction was proposed by Gentry and Silverberg [20]. The security relies on the Bilinear Diffie-Hellman assumption in the random oracle model. Subsequently, Boneh and Boyen [1] introduced HIBES in the selective-ID model without using random oracles. Boneh, Boyen and Goh [2] presented a selective-ID secure HIBE with constant size ciphertext. Gentry and Halevi [19] constructed a fully secure HIBES supporting polynomial hierarchy depth. In 2009, Waters [40] proposed a new framework, called Dual System

Encryption, for constructing fully secure IBES and HIBES. This approach has become a powerful tool for obtaining fully secure encryption schemes $[28,29]$. These plain HIBES are CPA-secure. The techniques in the previously reviewed conversions $[8,9,11]$ can be extended to achieve CCA2-secure HIBES with CPA-secure ones by adding one extra hierarchy to the underlying CPA-secure HIBES.

Generalized Identity-Based Encryption. Boneh and Hamburg [7] proposed a general framework for constructing IBES, named Generalized Identity-Based Encryption (GIBE), to incorporate different properties in IBE via a product rule. They also introduced an important instance of GIBE called Spatial Encryption (SE), showing that many GIBES are embedded in it, e.g., HIBE, inclusive IBE, co-inclusive IBE, in an identity-based like settings. HIBBE can also be derived from SE. However, the HIBBE derived from their SE only has selective and chosen-plaintext security. Very recently, Zhang et al. [43] suggested two fully secure and anonymous SE schemes, which not only obtain full security, but further protect the recipient identity privacy. Their constructions achieve CPA security and can be extended to CCA2 security, but also with the help of one-time signature schemes.

### 1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2, we review composite order bilinear groups and the assumptions used in our constructions. Section 3 formalizes HIBBE and its security definitions. We propose a secure HIBBES against Adaptively Chosen-Identity-Vector-Set and Chosen-Plaintext Attack in Section 4. We then introduce a compact transformation that converts our CPA-secure HIBBES into a CCA2-secure one in Section 5. We conclude the paper in Section 6.

## 2 Preliminaries

### 2.1 Composite Order Bilinear Groups

Composite order bilinear groups were first introduced in [6]. Let $\mathcal{G}$ be an algorithm which takes a security parameter $\lambda$ as input and outputs the description of a bilinear group, $\left(N, \mathbb{G}, \mathbb{G}_{T}, e\right)$, where $N=p_{1} p_{2} p_{3}$ is a composite integer with three distinct large prime factors $p_{1}, p_{2}$ and $p_{3}, \mathbb{G}$ and $\mathbb{G}_{T}$ are cyclic groups of order $N$, and a bilinear map $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{T}$ satisfying the following properties:

1. Bilinearity: for all $g, h \in \mathbb{G}$ and $a, b \in \mathbb{Z}_{N}, e\left(g^{a}, h^{b}\right)=e(g, h)^{a b}$;
2. Non-degeneracy: there exists at least an element $g \in \mathbb{G}$ such that $e(g, g)$ has order $N$ in $\mathbb{G}_{T}$;
3. Computability: There exists an efficient algorithm (in polynomial time with respect to $\lambda$ ) computing the bilinear pairing $e(u, v)$ for all $u, v \in \mathbb{G}$.

In addition to these properties, the three subgroups of order $p_{1}, p_{2}$ and $p_{3}$ in $\mathbb{G}$ (we respectively denote them by $\mathbb{G}_{p_{1}}, \mathbb{G}_{p_{2}}$ and $\mathbb{G}_{p_{3}}$ ) satisfy the orthogonality property:

$$
\text { For all } h_{i} \in \mathbb{G}_{p_{i}} \text { and } h_{j} \in \mathbb{G}_{p_{j}}, e\left(h_{i}, h_{j}\right)=1 \text { for } i \neq j
$$

This special property will be an essential tool in our constructions and the security proofs.

### 2.2 Assumptions in Composite Order Bilinear Groups

We will use three static assumptions to prove the security of our HIBBES. These three assumptions, which were first introduced by Lewko and Waters [28], hold if it is hard to find a nontrivial factor of $N$. Let $\mathcal{G}$ be a group generating algorithm that outputs a composite order bilinear group ( $N=$ $\left.p_{1} p_{2} p_{3}, \mathbb{G}, \mathbb{G}_{T}, e\right)$. For ease of description, we let $\mathbb{G}_{p_{i} p_{j}}$ denote the subgroup of order $p_{i} p_{j}$ in $\mathbb{G}$.

Let $g \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}$ be a random generator of $\mathbb{G}_{p_{1}}$ and $X_{3} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$ be a random element in $\mathbb{G}_{p_{3}}$. Assumption 1 is that it is hard to determine whether $T$ is a random element in $\mathbb{G}_{p_{1} p_{2}}$, or a random element in $\mathbb{G}_{p_{1}}$ given $D_{1}=\left(g, X_{3}\right)$ as an input. We define the advantage of an algorithm $\mathcal{A}$ that outputs $b \in\{0,1\}$ in solving the first assumption in $\mathbb{G}$ to be

$$
A d v 1_{\mathcal{A}}(\lambda)=\left|\operatorname{Pr}\left[\mathcal{A}\left(D_{1}, T \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1} p_{2}}\right)=1\right]-\operatorname{Pr}\left[\mathcal{A}\left(D_{1}, T \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}\right)=1\right]\right|
$$

Definition 1. Assumption 1 states that $\operatorname{Adv} 1_{\mathcal{A}}(\lambda)$ is negligible for all polynomial time algorithms $\mathcal{A}$.

Let $g \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}$ be a random generator of $\mathbb{G}_{p_{1}}$. Choose random elements $X_{1} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}, X_{2}, Y_{2} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{2}}$ and $X_{3}, Y_{3} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$. Assumption 2 is that given the input as $D_{2}=\left(g, X_{1} X_{2}, X_{3}, Y_{2} Y_{3}\right)$, it is hard to determine whether $T$ is a random element in $\mathbb{G}$ or a random element in $\mathbb{G}_{p_{1} p_{3}}$. We define the advantage of an algorithm $\mathcal{A}$ that outputs $b \in\{0,1\}$ in solving the second assumption in $\mathbb{G}$ to be

$$
\operatorname{Adv} 2_{\mathcal{A}}(\lambda)=\left|\operatorname{Pr}\left[\mathcal{A}\left(D_{2}, T \stackrel{R}{\leftarrow} \mathbb{G}\right)=1\right]-\operatorname{Pr}\left[\mathcal{A}\left(D_{2}, T \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1} p_{3}}\right)=1\right]\right|
$$

Definition 2. Assumption 2 states that $\operatorname{Adv} 2_{\mathcal{A}}(\lambda)$ is negligible for all polynomial time algorithms $\mathcal{A}$.
Similarly, let $g \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}$ be a random generator of $\mathbb{G}_{p_{1}}, X_{2}, Y_{2}, Z_{2} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{2}}$ be random elements in $\mathbb{G}_{p_{2}}, X_{3} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$ be a random element in $\mathbb{G}_{p_{3}}, \alpha, s \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ be random exponents chosen in $\mathbb{Z}_{N}$. Assumption 3 states that, given $D_{3}=\left(g, g^{\alpha} X_{2}, X_{3}, g^{s} Y_{2}, Z_{2}\right)$ as an input, it is hard to determine whether $T$ is $e(g, g)^{\alpha s}$, or a random element in $\mathbb{G}_{T}$. We define the advantage of an algorithm $\mathcal{A}$ that outputs $b \in\{0,1\}$ in solving the third assumption in $\mathbb{G}$ to be

$$
A d v 3_{\mathcal{A}}(\lambda)=\left|\operatorname{Pr}\left[\mathcal{A}\left(D_{3}, T \leftarrow e(g, g)^{\alpha s}\right)=1\right]-\left[\mathcal{A}\left(D_{3}, T \stackrel{R}{\leftarrow} \mathbb{G}_{T}\right)=1\right]\right|
$$

Definition 3. Assumption 3 states that $\operatorname{Adv} 3_{\mathcal{A}}(\lambda)$ is negligible for all polynomial time algorithms $\mathcal{A}$.

## 3 Syntax

### 3.1 Terminology and Notations

We introduce several notations to simplify the description of HIBBES. Table 1 summarizes these notations and their corresponding meanings.

Table 1. Notations

| Notation | Description | Notation | Description |
| :---: | :---: | :---: | :---: |
| $\lambda$ | Security Parameter | $P K$ | Public Key |
| $M S K$ | Master Key | $C T$ | Ciphertext |
| ID | Identity | ID | Identity Vector |
| I | Identity Vector Position | $S K_{\text {ID }}$ | Secret Key for Identity Vector |
| $\\|\mathbf{I D}\\|$ | Depth of ID | $S_{\text {ID }}$ | Identity Set Associated with ID |
| $\mathbf{V}$ | Identity Vector Set | $\mathbb{I}_{\mathbf{V}}$ | Identity Vector Set Position |
| $\\|\mathbf{V}\\|$ | Depth of $\mathbf{V}$ | $S_{\mathbf{V}}$ | Identity Set Associated with $\mathbf{V}$ |

We use $[a, b]$ to denote the integer set $\{a, a+1, \cdots, b\} .|S|$ denotes the cardinality of the set $S$. For an identity vector $\mathbf{I D}=\left(I D_{1}, I D_{2}, \cdots, I D_{d}\right)$, we define $\|\mathbf{I D}\|=d$ as the depth of ID and $S_{\text {ID }}=\left\{I D_{1}, \cdots, I D_{d}\right\}$ as the identity set assocaited with ID. The identity vector position of ID is defined by $\mathrm{I}_{\text {ID }}=\left\{i: I D_{i} \in S_{\text {ID }}\right\}$. Similarly, we define the maximal depth of an identity vector set as $\|\mathbf{V}\|=\max \{\|\mathbf{I D}\|: \mathbf{I D} \in \mathbf{V}\}$. The associated identity set $S_{\mathbf{V}}$ of $\mathbf{V}$ and the identity vector set position $\mathbb{I}_{\mathbf{V}}$ of $\mathbf{V}$ can be defined accordingly.

We slightly abuse the term prefix and define the prefix of an identity vector $\mathbf{I D}=\left(I D_{1}, \cdots, I D_{d}\right)$ as an identity vector set as $\operatorname{Pref}(\mathbf{I D})=\left\{\left(I D_{1}, \cdots, I D_{d^{\prime}}\right): d^{\prime} \leq d\right\}$. Clearly, $|\operatorname{Pref}(\mathbf{I D})|=\|\mathbf{I D}\|=d$. We similarly define the prefix of an identity vector set $\mathbf{V}$ as $\operatorname{Pref}(\mathbf{V})=\bigcup_{\mathbf{I D} \in \mathbf{V}} \operatorname{Pref}(\mathbf{I D})$.

In practice, a user may have more than one identity or parent node. In this case, we treat these users as different users with the same identity. Hence, without loss of generality, we assume that each user has a unique identity vector and can have at most one parent node.

For example, assume that the users are organized as in Figure 1. For the user whose identity vector is $\mathbf{I D}=\left(\mathrm{ID}_{1}, \mathrm{ID}_{3}\right)$, we have that $\|\mathbf{I D}\|=2, S_{\mathbf{I D}}=\left\{\mathrm{ID}_{1}, \mathrm{ID}_{3}\right\}$, and $\mathrm{I}_{\mathrm{ID}}=\{1,3\}$. The prefix of $\mathbf{I D}$ is $\operatorname{Pref}(\mathbf{I D})=\left\{\left(\mathrm{ID}_{1}\right),\left(\mathrm{ID}_{1}, \mathrm{ID}_{3}\right)\right\}$. Similarly, for the broadcast identity vector set $\mathbf{V}=\left\{\left(\mathrm{ID}_{1}, \mathrm{ID}_{3}\right),\left(\mathrm{ID}_{2}, \mathrm{ID}_{6}, \mathrm{ID}_{7}\right)\right\}$, we have that $\|\mathbf{V}\|=\max \{2,3\}=3$, the identity set associated with $\mathbf{V}$ is $S_{\mathbf{V}}=\left\{\mathrm{ID}_{1}, \mathrm{ID}_{3}, \mathrm{ID}_{2}, \mathrm{ID}_{6}, \mathrm{ID}_{7}\right\}$, and $\mathbb{I}_{\mathbf{V}}=\{1,3,2,6,7\}$. The prefix of $\mathbf{V}$ is

$$
\operatorname{Pref}(\mathbf{V})=\left\{\left(\mathrm{ID}_{1}\right),\left(\mathrm{ID}_{1}, \mathrm{ID}_{3}\right),\left(\mathrm{ID}_{2}\right),\left(\mathrm{ID}_{2}, \mathrm{ID}_{6}\right),\left(\mathrm{ID}_{2}, \mathrm{ID}_{6}, \mathrm{ID}_{7}\right)\right\}
$$



- PKG
(7) Nodes in Broadcast Vector Set V

Nodes Not in Broadcast Vector Set V

Fig. 1. A Typical Example of an HIBBES.

### 3.2 Hierarchical Identity-Based Broadcast Encryption

A ( $D, n$ )-HIBBES consists of five polynomial time algorithms: Setup, KeyGen, Delegate, Encrypt and Decrypt defined as follows:
$-\operatorname{Setup}(D, n, \lambda)$. Takes as inputs the maximal depth $D$ of the hierarchy, the maximal number $n$ of users, and the security parameter $\lambda$. It outputs a masker key $M S K$ and a public key $P K$.

- Encrypt $(P K, M, \mathbf{V})$. Takes as inputs the public key $P K$, a message $M$ in the message space $\mathcal{M}$, and a receiver identity vector set $\mathbf{V}$. It outputs the ciphertext $C T$ of the message $M$.
- KeyGen $(M S K$, ID). Takes as inputs the master key $M S K$ and an identity vector ID. It outputs a secret key $S K_{\text {ID }}$ for the user whose identity vector is ID.
- Delegate $\left(S K_{\mathbf{I D}^{\prime}}, I D\right)$. Takes as inputs a secret key of a user whose identity vector is $\mathbf{I D}^{\prime}$ of depth $d$ and an identity $I D$. It returns a secret key $S K_{\text {ID }}$ for the user whose identity vector is $\mathbf{I D}=\left(\mathbf{I D}^{\prime}, I D\right)$.
- Decrypt( $\left.\mathbf{V}, C T, S K_{\mathbf{I D}}\right)$. Takes as inputs a receiver identity vector set $\mathbf{V}$, a ciphertext $C T$ of a message $M$, and a secret key $S K_{\text {ID }}$ of a user whose identity vector is ID. If ID $\in \operatorname{Pref}(\mathbf{V})$, it returns $M$.

An HIBBES must satisfy the standard consistency constraint, namely for all $D \leq n \in \mathbb{N}$, all $(P P$, $M S K) \leftarrow \operatorname{Setup}(D, n, \lambda)$, all $S K_{\text {ID }} \leftarrow \operatorname{KeyGen}(M S K, \mathbf{I D})$ or $S K_{\text {ID }} \leftarrow \operatorname{Delegate}\left(S K_{\mathbf{I D}^{\prime}}, I D\right)$ with $\|\mathbf{I D}\| \leq D$, all $M \in \mathcal{M}$, and all $C T \leftarrow \operatorname{Encrypt}(P P, M, \mathbf{V})$ with $\|\mathbf{V}\| \leq D$ and $\left|S_{\mathbf{V}}\right| \leq n$, if $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$, then $\operatorname{Decrypt}\left(\mathbf{V}, C T, S K_{\mathbf{I D}}\right)=M$.

We define the security notion, named Ciphertext Indistinguishability against Adaptively Chosen-Identity-Vector-Set and Chosen-Ciphertext Attack (IND-CIVS-CCA2) for HIBBE. In this security model, the adversary is allowed to obtain the secret keys associated with any identity vectors ID of its choice and to issue decryption queries for its chosen ciphertexts, provided that the adversary does not query for the secret keys of its chosen receivers or their supervisors, or for the challenge ciphertext as one of its chosen messages. We require that even such an adversary cannot distinguish the encrypted messages of its choice.

IND-CIVS-CCA2 security is defined through a game played by an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Both of them are given the parameters $D, n$ and $\lambda$ as inputs.

- Setup. $\mathcal{C}$ runs Setup algorithm to obtain the public key $P K$ and gives it to $\mathcal{A}$.
- Phase 1. $\mathcal{A}$ adaptively issues two kinds of queries:
- Secret key query for an identity vector ID. $\mathcal{C}$ generates a secret key for ID and gives it to $\mathcal{A}$.
- Decryption query for the ciphertext $C T$ with a receiver identity vector set V. $\mathcal{C}$ responds by running algorithm KeyGen to generate a secret key $S K_{\text {ID }}$ for an identity vector ID satisfying $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$. It then decrypts the ciphertext $C T$ and returns the resulting message to $\mathcal{A}$.
- Challenge. When $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal-length messages $M_{0}$ and $M_{1}$ on which $\mathcal{A}$ wishes to be challenged. Also, $\mathcal{A}$ outputs a challenge identity vector set $\mathbf{V}^{*}$ which contains all the users that $\mathcal{A}$ wishes to attack. The identity vector set $\mathbf{V}^{*}$ should be such that for all the secret key queries for ID issued in Phase 1, ID $\notin \operatorname{Pref}\left(\mathbf{V}^{*}\right)$. $\mathcal{C}$ flips a random coin $b \stackrel{R}{\leftarrow}\{0,1\}$ and encrypts $M_{b}$ under the challenge identity vector set $\mathbf{V}^{*} . \mathcal{C}$ returns the challenge ciphertext $C T^{*}$ to $\mathcal{A}$.
- Phase 2. $\mathcal{A}$ further adaptively issues two kinds of queries:
- Secret key queries for identity vectors ID such that ID $\notin \operatorname{Pref}\left(\mathbf{V}^{*}\right)$.
- Decryption queries for ciphertexts $C T$ such that $C T \neq C T^{*}$.
$\mathcal{C}$ responds the same as in Phase 1.
- Guess. Finally, $\mathcal{A}$ outputs a guess $b^{\prime} \in\{0,1\}$ and wins in the game if $b=b^{\prime}$.

The advantage of such an $\mathcal{A}$ in attacking the $(D, n)$-HIBBES with security parameter $\lambda$ is defined as

$$
A d v_{\mathcal{A}, D, n}^{I N D-C I V S-C C A 2}(\lambda)=\left|\operatorname{Pr}\left[b^{\prime}=b\right]-\frac{1}{2}\right|
$$

Definition 4. A $(D, n)$-HIBBES is $\left(\tau, q, q_{d}, \epsilon\right)$-secure if for any $\tau$-time IND-CIVS-CCA2 adversary $\mathcal{A}$ that makes at most $q$ secret key queries and $q_{d}$ decryption queries, $\operatorname{Adv} v_{\mathcal{A}, D, n}^{I N D-C I V S-C C A 2}(\lambda)<\epsilon$.

As usual, we define Ciphertext Indistinguishability against Adaptively Chosen-Identity-VectorSet and Chosen-Plaintext Attack (IND-CIVS-CPA) for HIBBE as in the preceding game, with the constraint that $\mathcal{A}$ is not allowed to issue any decryption query. $\mathcal{A}$ is still able to adaptively issue secret key queries.

Definition 5. $A(D, n)$-HIBBES is $(\tau, q, \epsilon)$-secure if for any $\tau$-time IND-CIVS-CPA adversary $\mathcal{A}$ that makes at most $q$ secret key queries, we have that $A d v_{\mathcal{A}, D, n}^{I N D-C I V S-C P A}(\lambda)<\epsilon$.

It is challenging to achieve full (identity/identity-vector) security in BE and (H)IBE, some weaker security notions have been proposed to bridge security proofs or cater for special applications which require only moderate security levels. One useful security notion, called selective security, was first proposed by Canetti, Halevi, and Katz [10, 11] in IBES. In this notion, $\mathcal{A}$ should commits ahead of time to the challenge identity it will attack. Similar security notions can also be found in HIBES [1] and IBBES [14]. A counterpart security notion can be naturally defined in HIBBES, by requiring the adversary in HIBBE to submit a challenge identity vector set before seeing the public parameters.

Another useful security notion, named semi-static security, can also be extended in HIBBES. This security notion was first defined by Gentry and Waters [21] in BES. In this notion, $\mathcal{A}$ must first commit to a set $\bar{S}$ before the Setup phase. $\mathcal{A}$ cannot query for secret key of any user in $\bar{S}$, but it can attack any target set $S^{*} \subseteq \bar{S}$. This security notion is weaker than full security but stronger than selective security, since $\mathcal{A}$ can partly decide which set is allowed to query adaptively. In HIBBES, a similar security notion can be defined by requiring $\mathcal{A}$ to submit an identity vector set $\overline{\mathbf{V}}$ before the Setup phase and later allow $\mathcal{A}$ to challenge any identity vector set $\mathbf{V}^{*} \subseteq \operatorname{Pref}(\overline{\mathbf{V}})$. Recently, a practical HIBBES with moderate security result was proposed to meet this security notion [32].

## 4 IND-CIVS-CPA Secure HIBBE with Constant Size Ciphertext

In this section, we propose an IND-CIVS-CPA secure HIBBE with constant size ciphertext over composite order bilinear groups of order $N=p_{1} p_{2} p_{3}$. Our starting point is the Lewko-Waters fully secure HIBES [28] which was inspired by the Boneh-Boyen-Goh selectively secure HIBES [2]. To support broadcast, every user in our system, instead of every depth of hierarchy in $[2,28]$, is associated with a random element for blinding its own identity vector in its secret key. Since users' identities have been randomized by different elements, users cannot reveal any information about other users' secret keys from their own ones.

We realize the functionalities in $\mathbb{G}_{p_{1}}$, while randomizing secret keys in $\mathbb{G}_{p_{3}}$. The $\mathbb{G}_{p_{2}}$ space, called semi-functional space, is only used in security proofs.

### 4.1 Basic Construction

We first assume that the identity vectors at depth $k$ are vector elements in $\left(\mathbb{Z}_{N}\right)^{k}$. We later extend the construction to identity vectors over $\left(\{0,1\}^{*}\right)^{k}$ by first hashing each component $I D_{j} \in S_{\text {ID }}$ using a collision resistant hash function $H:\{0,1\}^{*} \rightarrow \mathbb{Z}_{N}$. We also assume that plaintexts are elements of $\mathbb{G}_{T}$. Similar to HIBES, we assume that users' positions in HIBBE are publicly known with the processing of KeyGen, Delegate, Encrypt and Decrypt. Our ( $D, n$ )-HIBBES works as follows.
$\operatorname{Setup}(D, n, \lambda)$. Run $\left(N, \mathbb{G}, \mathbb{G}_{T}, e\right) \leftarrow \mathcal{G}\left(1^{\lambda}\right)$ to generate a composite integer $N=p_{1} p_{2} p_{3}$, two groups $\mathbb{G}, \mathbb{G}_{T}$ of order $N$, and a bilinear map $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{T}$. Then, select a random generator $g \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}$, two random elements $h \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}, X_{3} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$, and a random exponent $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$. Next, pick random elements $u_{i} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{1}}$ for all $i \in[1, n]$. The public key $P K$ includes the description of ( $\left.N, \mathbb{G}, \mathbb{G}_{T}, e\right)$, as well as

$$
\left(g, h, u_{1}, \cdots, u_{n}, X_{3}, e(g, g)^{\alpha}\right)
$$

The master key is $M S K \leftarrow g^{\alpha}$.
KeyGen $(M S K, \mathbf{I D})$. For an identity vector ID of depth $d \leq D$, the key generation algorithm picks a random exponent $r \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and two random elements $A_{0}, A_{1} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$. It then chooses random elements $U_{j} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$ for all $j \in[1, n] \backslash \mathrm{I}_{\text {ID }}$ and outputs

$$
S K_{\mathbf{I D}} \leftarrow\left(g^{\alpha}\left(h \cdot \prod_{i \in \mathrm{I}_{\mathrm{ID}}} u_{i}^{I D_{i}}\right)^{r} A_{0}, g^{r} A_{1},\left\{u_{j}^{r} U_{j}\right\}_{j \in[1, n] \backslash \mathrm{I}_{\mathrm{ID}}}\right)
$$

Delegate $\left(S K_{\mathbf{I D}^{\prime}}, I D\right)$. Given a secret key

$$
S K_{\mathbf{I D}^{\prime}}=\left(g^{\alpha}\left(h \cdot \prod_{i \in \mathrm{I}_{\mathrm{ID}^{\prime}}} u_{i}^{I D_{i}}\right)^{r^{\prime}} A_{0}^{\prime}, g^{r^{\prime}} A_{1}^{\prime},\left\{u_{j}^{r^{\prime}} U_{j}^{\prime}\right\}_{j \in[1, n] \backslash \mathrm{I}_{\mathbf{I D}^{\prime}}}\right)=\left(a_{0}, a_{1},\left\{b_{j}\right\}_{j \in[1, n] \backslash \mathrm{I}_{\mathbf{I D}^{\prime}}}\right)
$$

the delegation algorithm generates a secret key for $\mathbf{I D}=\left(\mathbf{I D}^{\prime}, I D\right)$ as follows. It picks a random exponent $t \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$, and also chooses two random elements $R_{0}, R_{1} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$. Next, for all $j \in[1, n] \backslash \mathrm{I}_{\mathbf{I D}}$, it chooses random elements $T_{j} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$. The algorithm outputs

$$
S K_{\mathbf{I D}}=\left(a_{0}\left(b_{i}^{I D}\right)_{i \in \mathrm{I}_{\mathbf{I D}} \backslash \mathrm{I}_{\mathbf{I D}^{\prime}}}\left(h \prod_{i \in \mathrm{I}_{\mathbf{I D}}} u_{i}^{I D_{i}}\right)^{t} R_{0}, a_{1} g^{t} R_{1},\left\{b_{j} u_{j}^{t} T_{j}\right\}_{j \in[1, n] \backslash \mathrm{I}_{\mathbf{I D}}}\right)
$$

Note that by implicitly setting $r=r^{\prime}+t \in \mathbb{Z}_{N}, A_{0}=A_{0}^{\prime} U_{i}^{\prime} R_{0} \in \mathbb{G}_{p_{3}}$ with $i \in \mathrm{I}_{\mathbf{I D}} \backslash \mathrm{I}_{\mathbf{I D}^{\prime}}, A_{1}=A_{1}^{\prime} R_{1} \in$ $\mathbb{G}_{p_{3}}$, and $U_{j}=U_{j}^{\prime} T_{j} \in \mathbb{G}_{p_{3}}$ for all $j \in[1, n] \backslash \mathrm{I}_{\mathbf{I D}}$, this secret key can be written under the form

$$
S K_{\mathbf{I D}} \leftarrow\left(g^{\alpha}\left(h \cdot \prod_{i \in \mathrm{I}_{\mathbf{I D}}} u_{i}^{I D_{i}}\right)^{r} A_{0}, g^{r} A_{1},\left\{u_{j}^{r} U_{j}\right\}_{j \in[1, n] \backslash \mathrm{I}_{\mathbf{I D}}}\right)
$$

which is well-formed as if it were generated by the KeyGen algorithm. Hence $S K_{\mathbf{I D}}$ is a properly distributed secret key for $\mathbf{I D}=\left(\mathbf{I D}^{\prime}, I D\right)$.
$\operatorname{Encrypt}(P P, M, \mathbf{V})$. For the receiver identity vector set $\mathbf{V}$ the encryption algorithm picks a random exponent $\beta \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and outputs the ciphertext

$$
C T=\left(C_{0}, C_{1}, C_{2}\right)=\left(g^{\beta},\left(h \cdot \prod_{i \in \mathbb{I}_{\mathbf{V}}} u_{i}^{I D_{i}}\right)^{\beta}, e(g, g)^{\alpha \beta} \cdot M\right)
$$

$\operatorname{Decrypt}\left(\mathbf{V}, C T, S K_{\mathbf{I D}}\right)$. Given the ciphertext $C T=\left(C_{0}, C_{1}, C_{2}\right)$, any user whose identity vector satisfies ID $\in \operatorname{Pref}(\mathbf{V})$ can use its corresponding secret key $S K_{\mathbf{I D}}=\left(a_{0}, a_{1},\left\{b_{j}\right\}_{j \in[1, n] \backslash \text { IID }}\right)$ to compute

$$
K=a_{0} \cdot \prod_{j \in \mathbb{I}_{\mathbf{V}} \backslash \mathrm{I}_{\mathbf{I D}}} b_{j}^{I D_{j}}
$$

Then it outputs the message by calculating $M=C_{2} \cdot e\left(C_{1}, a_{1}\right) / e\left(K, C_{0}\right)$.

Soundness. If the ciphertext $C T=\left(C_{0}, C_{1}, C_{2}\right)$ is well-formed, then we have

$$
K=a_{0} \cdot \prod_{j \in \mathbb{I}_{\mathbf{V}} \backslash \mathrm{I}_{\mathbf{I D}}} b_{j}^{I D_{j}}=g^{\alpha}\left(h \cdot \prod_{i \in \mathbb{I}_{\mathbf{V}}} u_{i}^{I D_{i}}\right)^{r} \cdot\left(A_{0} \prod_{j \in \mathbb{I}_{\mathbf{V}} \backslash \mathrm{I}_{\mathbf{I D}}} U_{j}\right)
$$

Note that all random elements in $\mathbb{G}_{p_{3}}$ can be cancelled in the pairing operations due to the orthogonality property. Therefore, for the blinding factor in $C_{2}$, the following equalities hold:

$$
\begin{aligned}
\frac{e\left(C_{1}, a_{1}\right)}{e\left(K, C_{0}\right)}= & \frac{e\left(\left(h \prod_{i \in \mathbb{I}_{\mathbf{V}}} u_{i}^{I D_{i}}\right)^{\beta}, g^{r} A_{1}\right)}{e\left(g^{\alpha}\left(h \prod_{i \in \mathbb{I}_{\mathbf{V}}} u_{i}^{I D_{i}}\right)^{r}\left(A_{0} \prod_{j \in \mathbb{I}_{\mathbf{V}} \backslash \mathrm{I}_{\mathbf{I D}}} U_{j}\right), g^{\beta}\right)} \\
= & \frac{e\left(\left(h \cdot \prod_{i \in \mathbb{I}_{\mathbf{V}}} u_{i}^{I D_{i}}\right)^{\beta}, g^{r}\right)}{e\left(g^{\alpha}, g^{\beta}\right) \cdot e\left(h \cdot\left(\prod_{i \in \mathbb{I}_{\mathbf{V}}} u_{i}^{I D_{i}}\right)^{r}, g^{\beta}\right)}=\frac{1}{e(g, g)^{\alpha \beta}}
\end{aligned}
$$

It follows that

$$
C_{2} \cdot \frac{e\left(C_{1}, a_{1}\right)}{e\left(K, C_{0}\right)}=M \cdot \frac{e(g, g)^{\alpha \beta}}{e(g, g)^{\alpha \beta}}=M
$$

### 4.2 Security Analysis

The security of our scheme is guaranteed by the following Theorem. In a high level, the proof of our HIBBES follows the proof framework of Lewko-Waters HIBES [28], with an extra effort to generate ciphertexts for supporting broadcast.
Theorem 1. Let $\mathbb{G}$ be a group (of composite order $N$ ) endowed with an efficient bilinear map. Our HIBBES is IND-CIVS-CPA secure if all the three assumptions defined in Definition 1, Definition 2 and Definition 3 hold in $\mathbb{G}$.

To prove the IND-CIVS-CPA security of our scheme, we apply the Dual System Encryption technique introduced by Waters [40] for obtaining adaptively secure IBES and HIBES. This technique has been shown to be a powerful tool for security proofs [28, 29]. In a Dual System Encryption system, the ciphertexts and keys can take one of two indistinguishable forms: normal form and semi-functional form . Normal keys can decrypt normal or semi-functional ciphertexts, and semi-functional ciphertexts can be decrypted by normal or semi-functional keys. Decryption will fail when one uses a semi-functional key to decrypt a semi-functional ciphertext. Since these two kinds of keys and ciphertexts are indistinguishable, the simulator can replace all normal ciphertexts and keys with semi-functional ones in the security game. When all ciphertexts and keys are semi-functional, $\mathcal{A}$ obtains no information about the challenge ciphertext as none of the given keys are useful to decrypt the challenge ciphertext.

We first need to define the semi-functional key and the semi-functional ciphertext. They will only be used in the security proof. Let $g_{2} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{2}}$ be a random generator of $\mathbb{G}_{p_{2}}$, the semi-functional ciphertext and the semi-functional key are defined as follows:

Semi-Functional Ciphertext. Run Encrypt to construct a normal ciphertext $C T=\left(C_{0}^{\prime}, C_{1}^{\prime}, C_{2}^{\prime}\right)$. Then, choose random exponents $x, y_{c} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and set

$$
C_{0}=C_{0}^{\prime}, C_{1}=C_{1}^{\prime} g_{2}^{x y_{c}}, C_{2}=C_{2}^{\prime} g_{2}^{x}
$$

Semi-Functional Key. For an identity vector ID, run KeyGen to generate its normal secret key

$$
S K=\left(a_{0}^{\prime}, a_{1}^{\prime},\left\{b_{j}^{\prime}\right\}_{j \in[1, n] \backslash \mathrm{I}_{\mathrm{ID}}}\right)
$$

Then, choose random exponents $\gamma, y_{k} \in \mathbb{G}_{N}, z_{j} \in \mathbb{G}_{N}$ for all $j \in[1, n] \backslash \mathrm{I}_{\text {ID }}$ and set

$$
a_{0}=a_{0}^{\prime} g_{2}^{\gamma}, a_{1}=a_{1}^{\prime} g_{2}^{\gamma y_{k}},\left\{b_{j}=b_{j}^{\prime} g_{2}^{\gamma z_{j}}\right\}_{j \in[1, n] \backslash \mathrm{I}_{\mathrm{ID}}}
$$

Decrypt will correctly output the message $M$ when decrypting a semi-functional ciphertext using a normal key or a semi-functional key since the added elements in $\mathbb{G}_{p_{2}}$ will be cancelled due to the orthogonality property. However, the blinding factor will be multiplied by the additional term $e\left(g_{2}, g_{2}\right)^{x \gamma\left(y_{k}-y_{c}\right)}$ when trying to decrypt the semi-functional ciphertext using a semi-functional key, unless $y_{k}=y_{c}$ with probability $\frac{1}{N}$. In this case, we call the key a nominally semi-functional key. In the semi-functional secret key, the exponent $y_{k}$ used for blinding the second component $a_{1}$ and the exponents $z_{j}$ used for blinding the third component $a_{2}$ are chosen randomly and only appear at most twice in the security game. Therefore, from $\mathcal{A}$ 's view the components in $\mathbb{G}_{p_{2}}$ for the semi-functional secret keys look random so that it does not helpful for $\mathcal{A}$ to distinguish the semi-functional secret key from a normal one, except with negligible probability $\frac{1}{N}$ when nominally semi-functional secret keys is coincidentally generated.

We prove security by using a sequence of games:

- Game Real . It is the real security game.
- Game $_{\text {Restricted. }}$. It is identical with Game $_{\text {Real }}$, except that in Phase 2, $\mathcal{A}$ cannot ask for identity vectors ID $=\left(I D_{1}, \cdots, I D_{d}\right)$ satisfying $\exists \mathbf{I D}^{*}=\left(I D_{1}^{*}, \cdots, I D_{d^{\prime}}^{*}\right) \in \operatorname{Pref}\left(\mathbf{V}^{*}\right)$ with $d^{\prime} \leq d$, s.t. $\forall i \in\left[1, d^{\prime}\right], I D_{i}=I D_{i}^{*} \bmod p_{2}$, where $\mathbf{V}^{*}$ is the challenge identity vector set.
- Game $_{\mathbf{k}}$. Suppose that $\mathcal{A}$ can make $q$ secret key queries in Phase 1 and Phase 2. This game is identical with the $\mathbf{G a m e}_{\text {Restricted }}$, except that the challenge ciphertext is semi-functional and the first $k$ keys are semi-functional, while the rest of the keys are normal. We note that in Game $\mathbf{G}_{\mathbf{0}}$, only the challenge ciphertext is semi-functional; in $\mathbf{G a m e}_{\mathbf{q}}$, the challenge ciphertext and all secret keys are semi-functional.
- Game $_{\text {Final }}$. It is the same as $\mathbf{G a m e}_{q}$, except that the challenge ciphertext is a semi-functional encryption of a random message in $\mathbb{G}_{T}$, not one of the messages given by $\mathcal{A}$.

Given a security parameter $\lambda$, we respectively represent the advantages of winning in the games Game $_{\text {Real }}$, Game $_{\text {Restricted }}$, Game $_{\mathbf{k}}$ and $\mathbf{G a m e}_{\text {Final }}$ by $\operatorname{Adv} v_{\text {Real }}^{\mathrm{CPA}}(\lambda), \operatorname{Adv} v_{\text {Restricted }}^{\mathrm{CPA}}(\lambda), \operatorname{Adv} v_{k}^{\mathrm{CPA}}(\lambda)$ and $A d v_{\text {Final }}^{\mathrm{CPA}}(\lambda)$. We show that these games are indistinguishable in the following four lemmas.

Lemma 1. Suppose Assumption 2 defined in Definition 2 holds. Then there is no polynomial time algorithm that can distinguish $\boldsymbol{G a m e}_{\text {Real }}$ from $\boldsymbol{G a m e}_{\boldsymbol{R e s t r i c t e d}}$ with non-negligible advantage.

Proof. If there exists an adversary $\mathcal{A}$ that can distinguish $\mathbf{G a m e}_{\text {Real }}$ from $\mathbf{G a m e}_{\text {Restricted }}$ with advantage $\epsilon_{R}$, then by the definition of Game $_{\text {Restricted }} \mathcal{A}$ can issue a secret key query for the identity vector $\mathbf{I D}=\left(I D_{1}, \cdots, I D_{d}\right)$ from others satisfying that

$$
\exists \mathbf{I} \mathbf{D}^{*}=\left(I D_{1}^{*}, \cdots, I D_{d^{\prime}}^{*}\right) \in \operatorname{Pref}\left(\mathbf{V}^{*}\right) \text { with } d^{\prime} \leq d \text {, s.t. } \forall i \in\left[1, d^{\prime}\right], I D_{i}=I D_{i}^{*} \bmod p_{2}
$$

Then a factor of $N$ can be extracted by computing $\operatorname{gcd}\left(I D_{i}-I D_{i}^{*}, N\right)$, from which we can build a similar algorithm described in the proof of Lemma 5 in [28] that can refute the second assumption with advantage $\operatorname{Adv} 2_{\mathcal{B}}(\lambda) \geq \epsilon_{R} / 2$. We omit the details here for avoiding repetition.

Compared with $\mathbf{G a m e}_{\text {Restricted }}$, in $\mathbf{G a m e}_{\mathbf{0}}$ the challenge ciphertext is replaced with a semifunctional one. Since $\mathcal{A}$ does not know the factor of $N=p_{1} p_{2} p_{3}$, it cannot determine whether the components of the challenge ciphertext are in $\mathbb{G}_{p_{1}}$ or in $\mathbb{G}_{p_{1} p_{2}}$. Hence $\mathcal{A}$ is unable to know of which form the given challenge ciphertext is. This implies indistinguishability between $\mathbf{G a m e}_{\text {Restricted }}$ and Game $_{\mathbf{0}}$. Formally, we have the following Lemma.

Lemma 2. Suppose Assumption 1 defined in Definition 1 holds. Then there is no polynomial time


Proof. Suppose that there exists an adversary $\mathcal{A}$ that can distinguish Game $_{\text {Restricted }}$ from Game $_{\mathbf{0}}$ with advantage $\epsilon_{0}$. Then we can construct an algorithm $\mathcal{B}$ that can refute Assumption 1 with advantage $\operatorname{Adv} 1_{\mathcal{B}}(\lambda) \geq \epsilon_{0}$. The input of $\mathcal{B}$ is the challenge tuple $\left(g, X_{3}, T\right)$ of Assumption 1. $\mathcal{B}$ needs to determine whether $T$ is in $\mathbb{G}_{p_{1}}$ or in $\mathbb{G}_{p_{1} p_{2}} . \mathcal{B}$ sets the public key as follows. It randomly chooses $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and
$\gamma_{i} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $i \in[0, n]$. Then, it sets $h \leftarrow g^{\gamma_{0}}$ and $u_{i} \leftarrow g^{\gamma_{i}}$ for all $i \in[1, n]$. Finally, $\mathcal{B}$ gives the public key $P K \leftarrow\left(g, h, u_{1}, \cdots, u_{n}, X_{3}, e(g, g)^{\alpha}\right)$ to $\mathcal{A}$. It keeps the master key $M S K \leftarrow g^{\alpha}$ to itself.

Assume that $\mathcal{A}$ issues a secret key query for the identity vector ID $=\left(I D_{1}, \cdots I D_{d}\right) . \mathcal{B}$ chooses random elements $r, w_{0}, w_{1} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and $v_{j} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $j \in[1, n] \backslash \mathrm{I}$, where $\mathrm{I}=\left\{i: I D_{i} \in S_{\mathrm{ID}}\right\}$. Then $\mathcal{B}$ returns a normal key

$$
S K_{\mathrm{ID}} \leftarrow\left(g^{\alpha}\left(h \cdot \prod_{i \in \mathrm{I}} u_{i}^{I D_{i}}\right)^{r} X_{3}^{w_{0}}, g^{r} X_{3}^{w_{1}},\left\{u_{j}^{r} X_{3}^{v_{j}}\right\}_{j \in[1, n] \backslash \mathrm{I}}\right)
$$

When $\mathcal{A}$ decides that the Challenge phase starts, it outputs two equal-length messages $M_{0}, M_{1} \in$ $\mathbb{G}_{T}$, together with a challenge identity vector set $\mathbf{V}^{*} . \mathcal{B}$ flips a random coin $b \stackrel{R}{\leftarrow}\{0,1\}$, and returns the challenge ciphertext

$$
C T^{*} \leftarrow\left(C_{0}^{*}, C_{1}^{*}, C_{2}^{*}\right) \leftarrow\left(T, T^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}}, M_{b} \cdot e(g, T)^{\alpha}\right)
$$

where $\mathbb{I}^{*}=\left\{i: I D_{i}^{*} \in S_{\mathbf{V}^{*}}\right\}$.
$\mathcal{A}$ outputs a guess that it is in Game $_{\text {Restricted }}$ or in Game $_{\mathbf{0}} . \mathcal{B}$ guesses $T \in \mathbb{G}_{p_{1}}$ if $\mathcal{A}$ decides it is in Game $_{\text {Restricted }}$. Otherwise, $\mathcal{B}$ outputs $T \in \mathbb{G}_{p_{1} p_{2}}$.

If $T \in \mathbb{G}_{p_{1}}$, this is a normal ciphertext by implicitly setting $T \leftarrow g^{\beta}$. Hence, $\mathcal{B}$ is simulating Game $_{\text {Restricted }}$. Otherwise, if $T \in \mathbb{G}_{p_{1} p_{2}}$, all components in this ciphertext contain elements in subgroup $\mathbb{G}_{p_{2}}$, thus it is a semi-functional ciphertext. In this case, $\mathcal{B}$ is simulating Game $_{\mathbf{0}}$. If $\mathcal{A}$ has
 with advantage $\operatorname{Adv} 1_{\mathcal{B}}(\lambda) \geq \epsilon_{0}$.

Similarly, $\mathbf{G a m e}_{\mathbf{k}-\mathbf{1}}$ and $\mathbf{G a m e}_{\mathbf{k}}$ are two indistinguishable games. The way to determine whether the $k^{\text {th }}$ queried key is normal or semi-functional is to determine whether the key components are in $\mathbb{G}_{p_{1} p_{3}}$ or in $\mathbb{G}_{N}$. This is computationally difficult without factoring $N=p_{1} p_{2} p_{3}$. Hence, we have the following Lemma.

Lemma 3. Suppose Assumption 2 defined in Definition 2 holds. Then there is no polynomial time algorithm that can distinguish $\boldsymbol{G a m e}_{k-1}$ from $\boldsymbol{G a m e}_{k}$ with non-negligible advantage.

Proof. Suppose there exists an adversary $\mathcal{A}$ that can distinguish $\mathbf{G a m e}_{\mathbf{k} \mathbf{- 1}}$ from $\mathbf{G a m e}_{\mathbf{k}}$ with advantage $\epsilon_{k}$. Then we can construct an algorithm $\mathcal{B}$ that can refute Assumption 2 with advantage $\operatorname{Adv} 2_{\mathcal{B}}(\lambda) \geq \epsilon_{k}$. The input of $\mathcal{B}$ is the challenge tuple ( $g, X_{1} X_{2}, X_{3}, Y_{2} Y_{3}, T$ ) of Assumption 2. $\mathcal{B}$ has to answer $T$ is in $\mathbb{G}_{N}$ or in $\mathbb{G}_{p_{1} p_{3}}$.
$\mathcal{B}$ runs exactly the same as Setup in the proof of Lemma 2. The public key can be published as $P K \leftarrow\left(g, h, u_{1}, \cdots, u_{n}, X_{3}, e(g, g)^{\alpha}\right)$ with $g \leftarrow g, h \leftarrow g^{\gamma_{0}}$ and $u_{i} \leftarrow g^{\gamma_{i}}$ for all $i \in[1, n]$. The master key is $M S K \leftarrow g^{\alpha}$ that is kept secret to $\mathcal{B}$.

When receiving the $\ell^{\text {th }}$ secret key query for identity vector $\mathbf{I D}=\left(I D_{1}, \cdots I D_{d}\right)$ with $\ell<k, \mathcal{B}$ creates a semi-functional key to response to the query. Denote $\mathrm{I}=\left\{i: I D_{i} \in S_{\mathbf{I D}}\right\}$. $\mathcal{B}$ chooses random elements $r, w_{0}, w_{1} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and $v_{j} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $j \in[1, n] \backslash$ I. Then it returns the secret key

$$
S K_{\mathbf{I D}} \leftarrow\left(g^{\alpha}\left(h \cdot \prod_{i \in \mathrm{I}} u_{i}^{I D_{i}}\right)^{r}\left(Y_{2} Y_{3}\right)^{w_{0}}, g^{r}\left(Y_{2} Y_{3}\right)^{w_{1}},\left\{u_{j}^{r}\left(Y_{2} Y_{3}\right)^{v_{j}}\right\}_{j \in[1, n] \backslash \mathrm{I}}\right)
$$

This is a well-formed semi-functional key obtained by implicitly setting $g_{2}^{\gamma}=Y_{2}^{w_{0}}$ and $y_{k}=w_{1} / w_{0}$.
If $\mathcal{A}$ issues the $\ell^{\text {th }}$ secret key query for $k<l \leq q, \mathcal{B}$ calls the usual key generation algorithm to generate a normal secret key and returns it to $\mathcal{A}$.

When $\mathcal{A}$ issues the $k^{\text {th }}$ secret key query for identity vector ID with $\mathrm{I}=\left\{i: I D_{i} \in S_{\text {ID }}\right\}, \mathcal{B}$ chooses random exponents $w_{0} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and $v_{j} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $j \in[1, n] \backslash$ I. It then outputs

$$
S K_{\mathrm{ID}} \leftarrow\left(g^{\alpha} T^{\gamma_{0}+\sum_{i \in \mathrm{I}} I D_{i} \cdot \gamma_{i}} X_{3}^{w_{0}}, T,\left\{T^{\gamma_{j}} X_{3}^{v_{j}}\right\}_{j \in[1, n] \backslash \mathrm{I}}\right)
$$

If $T \in \mathbb{G}_{p_{1} p_{3}}$, then all components in this secret key are in $\mathbb{G}_{p_{1} p_{3}}$. Hence it is a normal secret key. Otherwise, it is a semi-functional key by implicitly setting $y_{k}=\gamma_{0}+\sum_{i \in \mathrm{I}} I D_{i} \cdot \gamma_{i}$.

In Challenge phase, $\mathcal{B}$ receives two equal-length messages $M_{0}, M_{1} \in \mathbb{G}_{T}$ and a challenge identity vector set $\mathbf{V}^{*}$ from $\mathcal{A}$. It chooses a random bit $b \stackrel{R}{\leftarrow}\{0,1\}$ and returns

$$
C T^{*} \leftarrow\left(C_{0}^{*}, C_{1}^{*}, C_{2}^{*}\right) \leftarrow\left(X_{1} X_{2},\left(X_{1} X_{2}\right)^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}}, M_{b} \cdot e\left(g, X_{1} X_{2}\right)^{\alpha}\right)
$$

to $\mathcal{A}$, where $\mathbb{I}^{*}=\left\{i: I D_{i}^{*} \in S_{\mathbf{V}^{*}}\right\}$.
Note that this ciphertext is semi-functional with $y_{c}=\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}$. Since from Game $\mathbf{G e s t r i c t e d}$, the identity vector associating with the $k^{\text {th }}$ secret key is not a prefix of the challenge receiver identity vector set modulo $p_{2}, y_{c}$ and $y_{k}$ will seem randomly distributed to $\mathcal{A}$ so that the relationship between $y_{c}$ and $y_{k}$ offers no help for $\mathcal{A}$ to distinguish the two games.

Although hidden from $\mathcal{A}$, the relationship between $y_{c}$ and $y_{k}$ is important: it prevents $\mathcal{B}$ from testing if the $k^{\text {th }}$ secret key is semi-functional by generating a semi-functional ciphertext for any identity vector set $\mathbf{V}$ with $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$ and decrypts it using the $k^{\text {th }}$ key. Indeed, $\mathcal{B}$ only can generate a nominally semi-functional key for the $k^{\text {th }}$ key query for ID. Note that $y_{k}+\sum_{i \in \mathbb{I} \backslash \mathrm{I}} I D_{i} \cdot \gamma_{i}=y_{c}$, where $\mathrm{I}=\left\{i: I D_{i} \in S_{\mathbf{I D}}\right\}$ and $\mathbb{I}=\left\{i: I D_{i} \in S_{\mathbf{V}}\right\}$. Hence, if $\mathcal{B}$ tries to do that, then decryption will always work, even when the $k^{t h}$ key is semi-functional. So, using this method, $\mathcal{B}$ cannot test whether the $k^{\text {th }}$ key for identity vector ID is semi-functional or not without $\mathcal{A}$ 's help. Note that this is the only case the nominally semi-functional secret key is used. For other queried secret keys, the exponents used in the subgroup $\mathbb{G}_{p_{2}}$ are randomly chosen so that the secret keys are randomly blinded by the elements in $\mathbb{G}_{p_{2}}$ and helpless for $\mathcal{A}$ to win the security game.
$\mathcal{B}$ finally outputs $T \in \mathbb{G}_{p_{1} p_{3}}$ if $\mathcal{A}$ outputs that it is in $\mathbf{G a m e}_{\mathbf{k - 1}}$, or outputs $T \in \mathbb{G}_{N}$ if $\mathcal{A}$ answers that it is in Game $_{\mathbf{k}}$.

If $T \in \mathbb{G}_{p_{1} p_{3}}$, all components in the $k^{\text {th }}$ secret key generated by $\mathcal{B}$ are in $\mathbb{G}_{p_{1} p_{3}}$. Hence it is a normal secret key. In this case, $\mathcal{B}$ is simulating Game $_{\mathbf{k}-\mathbf{1}}$. Otherwise, if $T \in \mathbb{G}_{N}$, then the $k^{\text {th }}$ secret key is semi-functional. In this case, $\mathcal{B}$ is simulating $\mathbf{G a m e}_{\mathbf{k}}$. If $\mathcal{A}$ has advantage $\epsilon_{k}$ in distinguishing these two games, $\mathcal{B}$ can also distinguish $T \in \mathbb{G}_{p_{1} p_{3}}$ from $T \in \mathbb{G}_{N}$ with advantage $A d v_{\mathcal{B}}(\lambda) \geq \epsilon_{k}$.
Lemma 4. Suppose Assumption 3 defined in Definition 3 holds. Then there is no polynomial time algorithm that can distinguish Game $_{\boldsymbol{q}}$ from $\boldsymbol{G a m e}_{\text {Final }}$ with non-negligible advantage.

Proof. Suppose that there exists an adversary $\mathcal{A}$ that can distinguish $\mathbf{G a m e}_{\mathbf{q}}$ from $\mathbf{G a m e}_{\mathbf{F i n a l}}$ with advantage $\epsilon_{F}$. By invoking $\mathcal{A}$ as a blackbox, we build an algorithm $\mathcal{B}$ refuting the third assumption with advantage $\operatorname{Adv} 3_{\mathcal{B}}(\lambda) \geq \epsilon_{F}$. $\mathcal{B}$ is given the challenge tuple ( $g, g^{\alpha} X_{2}, X_{3}, g^{s} Y_{2}, Z_{2}, T$ ) and is required to answer whether $T$ is $e(g, g)^{\alpha s}$ or a random element in $\mathbb{G}_{T} . \mathcal{B}$ randomly chooses $\gamma_{i} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $i \in[0, n]$ and sets the public key

$$
P K \leftarrow\left(g=g, h=g^{\gamma_{0}}, u_{1}=g^{\gamma_{1}}, \cdots, u_{n}=g^{\gamma_{n}}, X_{3}, e(g, g)^{\alpha}=e\left(g^{\alpha} X_{2}, g\right)\right)
$$

When $\mathcal{A}$ requests a secret key for an identity vector $\mathbf{I D}, \mathcal{B}$ chooses random exponents $w_{0}, w_{1}, t_{0}, t_{1} \stackrel{R}{\leftarrow}$ $\mathbb{Z}_{N}$ and $v_{j}, z_{j} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $j \in[1, n] \backslash \mathrm{I}$, where $\mathrm{I}=\left\{i: I D_{i} \in S_{\mathrm{ID}}\right\}$. It outputs

$$
S K_{\mathrm{ID}} \leftarrow\left(g^{\alpha} X_{2}\left(h \cdot \prod_{i \in \mathrm{I}} u_{i}^{I D_{i}}\right)^{r} Z_{2}^{t_{0}} X_{3}^{w_{0}}, g^{r} Z_{2}^{t_{1}} X_{3}^{w_{1}},\left\{u_{j}^{r} Z_{2}^{z_{j}} X_{3}^{v_{j}}\right\}_{j \in[1, n] \backslash \mathrm{I}}\right)
$$

Note that this secret key is semi-functional with $g_{2}^{\gamma}=Z_{2}^{t_{0}}$ and $y_{k}=t_{1} / t_{0}$.
In the challenge phase, $\mathcal{A}$ outputs two equal-length messages $M_{0}, M_{1} \in \mathbb{G}_{T}$, and a challenge identity vector set $\mathbf{V}^{*}$. Denote $\mathbb{I}^{*}=\left\{i: I D_{i}^{*} \in S_{\mathbf{V}^{*}}\right\} . \mathcal{B}$ chooses a random bit $b \stackrel{R}{\leftarrow}\{0,1\}$ and outputs the resulting semi-functional ciphertext

$$
C T^{*} \leftarrow\left(C_{0}^{*}, C_{1}^{*}, C_{2}^{*}\right) \leftarrow\left(g^{s} Y_{2},\left(g^{s} Y_{2}\right)^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}}, M_{b} \cdot T\right)
$$

Eventually, if $\mathcal{A}$ guesses that it is in $\mathbf{G a m e}_{\mathbf{q}}, \mathcal{B}$ outputs $T \leftarrow e(g, g)^{\alpha s}$. Otherwise, $\mathcal{B}$ outputs $T \stackrel{R}{\leftarrow} \mathbb{G}_{T}$ when $\mathcal{A}$ answers that it is in Game $_{\text {Final }}$.

If $T \leftarrow e(g, g)^{\alpha s}$, then $\mathcal{B}$ is simulating $\mathbf{G a m e}_{\mathbf{q}}$ since $C T^{*}$ is a semi-functional ciphertext of the message $M_{b}$. If $T \stackrel{R}{\leftarrow} \mathbb{G}_{T}$, then $C T^{*}$ is a semi-functional ciphertext of a random message that is independent of $M_{b}$. In this case, $\mathcal{B}$ is simulating Game $_{\text {Final }}$. Hence, if $\mathcal{A}$ has advantage $\epsilon_{F}$ in distinguishing these two games, then $\mathcal{B}$ has advantage $\operatorname{Adv} 3_{\mathcal{B}}(\lambda) \geq \epsilon_{F}$ in distinguishing the distribution of $T$.

Since all keys and ciphertexts are semi-functional in $\mathbf{G a m e}_{\mathbf{q}}, \mathcal{A}$ can get no information about the challenge ciphertext since none of the given keys are useful to decrypt it. Therefore, $\mathcal{A}$ cannot notice that the challenge ciphertext has been replaced by a random element. This implies the indistinguishability between $\mathbf{G a m e}_{\mathbf{q}}$ and $\mathbf{G a m e}_{\text {Final }}$.

With the above lemmas, these games are indistinguishable and in the final game the encrypted message is information-theoretically hidden from $\mathcal{A}$. Therefore, the proof of Theorem 1 follows.

Proof. If the three assumptions hold, then for all polynomial time adversaries $\mathcal{A}, \operatorname{Adv} 1_{\mathcal{A}}(\lambda), \operatorname{Adv} 2_{\mathcal{A}}(\lambda)$, and $\operatorname{Adv} 3_{\mathcal{A}}(\lambda)$ are all negligible probability. In Game Final , the ciphertext has been replaced with a random element of $\mathbb{G}_{T}$. The value of $b$ chosen by the challenger is information-theoretically hidden from $\mathcal{A}$. By applying the Lemma 1, Lemma 2, Lemma 3 and Lemma 4, we have that

$$
\begin{aligned}
\left|A d v_{\text {Real }}^{\mathrm{CPA}}(\lambda)\right| & \leq\left|A d v_{\text {Real }}^{\mathrm{CPA}}(\lambda)-A d v_{\text {Restricted }}^{\mathrm{CPA}}(\lambda)+A d v_{\text {Restricted }}^{\mathrm{CPA}}(\lambda)-\cdots-A d v_{\text {Final }}^{\mathrm{CPA}}(\lambda)+A d v_{\mathrm{Final}}^{\mathrm{CPA}}(\lambda)\right| \\
& \leq\left|A d v_{\text {Real }}^{\mathrm{CPA}}(\lambda)-A d v_{\text {Restricted }}^{\mathrm{CPA}}(\lambda)\right|+\cdots+\left|A d v_{q}^{\mathrm{CPA}}(\lambda)-A d v_{\text {Final }}^{\mathrm{CPA}}(\lambda)\right|+\left|A d v_{\text {Final }}^{\mathrm{CPA}}(\lambda)\right| \\
& \leq A d v 1_{\mathcal{A}}(\lambda)+(q+2) \cdot A d v 2_{\mathcal{A}}(\lambda)+A d v 3_{\mathcal{A}}(\lambda)
\end{aligned}
$$

Therefore, there is no polynomial time adversary that can break our HIBBES with non-negligible advantage. This completes the proof of Theorem 1.

## 5 Compact IND-CIVS-CCA2 HIBBE with Short Ciphertexts

### 5.1 Basic Ideas

In this section, we construct an IND-CIVS-CCA2 secure ( $D, n$ )-HIBBES from our IND-CIVS-CPA secure ( $D, n+1$ )-HIBBES. We first provide an overview of the conversion. We add one "dummy user" with an on-the-fly "identity" to the system. This dummy user is at depth 1, i.e., a child of the PKG. No one is allowed to obtain the secret key for the dummy user. It will be used just for the ciphertext validity test. When encrypting a message $M$, the encryption algorithm first creates the ciphertext components $C_{0}$ and $C_{2}$, which are independent of the receiver's identity vector set. Then, the algorithm hashes these two elements using a collision resistant hash function, and assigns it as the on-the-fly "identity" of the dummy user. Finally, we compute the ciphertext component $C_{1}$, as in the encryption algorithm of CPA-secure scheme. We show that there is an efficient algorithm to verify whether the resulting ciphertext is valid or not. In one word, the ciphertext validity test can be done publicly, since the test only involves the ciphertext $C T$ and the public key $P K$.

This technique is inspired by the Boyen-Mei-Waters technique [9], which applies to Waters' adaptively secure IBE [39] and Boneh-Boyen selective-ID secure IBE [1] to obtain CCA2-secure public key cryptosystems. Boyen et al. remarked that their technique can be extended to achieve CCA2-secure HIBES from some CPA-secure HIBES by adding one extra hierarchy to the underlying HIBES. Instead of introducing one extra hierarchy of users to our HIBBE, we just add one extra dummy user at the first level by exploiting the broadcasting feature to enforce ciphertext validation test. In this way, CCA2 security is achieved only at a marginal cost of one extra user.

### 5.2 The Resulting Construction

For simple description, we label the previous HIBBES as HIBBE CPA with algorithms Setup CPA $^{\text {, }}$ KeyGen $_{\text {CPA }}$, Delegate ${ }_{\text {CPA }}$, Encrypt ${ }_{\text {CPA }}$, and Decrypt CPA $^{\text {. Our CCA2-secure HIBBES is denoted }}$ by $\operatorname{HIBBE}_{\mathrm{CCA} 2}$. Similar to $\mathrm{HIBBE}_{\mathrm{CPA}}$, we assume that the identity vectors ID $=\left(I D_{1}, \cdots, I D_{k}\right)$ at depth $k$ are vector elements in $\left(\mathbb{Z}_{N}\right)^{k}$, and messages to be encrypted are elements in $\mathbb{G}_{T}$. Our resulting scheme works as follows:
$\operatorname{Setup}(D, n, \lambda)$. The system first runs $\operatorname{Setup}_{\mathrm{CPA}}(D, n+1, \lambda)$ to generate the public key

$$
P K \leftarrow\left(g, h, u_{1}, \cdots, u_{n}, u_{n+1}, X_{3}, e(g, g)^{\alpha}\right)
$$

and the master key $M S K \leftarrow g^{\alpha}$. A collision resistant hash function $H: \mathbb{G} \times \mathbb{G}_{T} \rightarrow \mathbb{Z}_{N}$ is also included in the public key. We stress that the dummy user, associated with parameter $u_{n+1}$, is at depth 1 and no one is allowed to obtain its corresponding secret key.

KeyGen and Delegate. These two algorithms are identical to KeyGen CPA and Delegate ${ }_{\text {CPA }}$.
$\operatorname{Encrypt}(P K, M, \mathbf{V})$. For a receiver identity vector set $\mathbf{V}$, denote $\mathbb{I}=\left\{i: I D_{i} \in S_{\mathbf{V}}\right\}$. The encryption algorithm first picks a random $\beta \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and computes

$$
\left(C_{0}, C_{2}\right) \leftarrow\left(g^{\beta}, e(g, g)^{\alpha \beta} \cdot M\right)
$$

Then, the algorithm computes $I D_{n+1} \leftarrow H\left(C_{0}, C_{2}\right) \in \mathbb{Z}_{N}$ and constructs $C_{1}$ as

$$
C_{1} \leftarrow\left(h \cdot u_{n+1}^{I D_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_{i}^{I D_{i}}\right)^{\beta}
$$

The algorithm finally outputs the ciphertext as $C T \leftarrow\left(C_{0}, C_{1}, C_{2}\right)$. Note that it is a valid HIBBE CPA ciphertext for the receiver identity vector set $\mathbf{V} \cup\left\{\left(I D_{n+1}\right)\right\}$.
$\operatorname{Decrypt}\left(\mathbf{V}, C T, S K_{\mathbf{I D}}\right)$. Suppose the secret key for the user associated with identity vector ID is

$$
S K_{\mathbf{I D}}=\left(a_{0}, a_{1},\left\{b_{j}\right\}_{j \in[1, n+1] \backslash \mathrm{I}}\right)
$$

where $\mathrm{I}=\left\{i: I D_{i} \in S_{\mathbf{I D}}\right\}$. Denote $\mathbb{I}=\left\{i: I D_{i} \in S_{\mathbf{V}}\right\}$. Before decrypting the ciphertext $C T=$ $\left(C_{0}, C_{1}, C_{2}\right)$, the decryption algorithm needs to first verify whether the ciphertext is legitimate. It does this by randomly choosing elements $Z_{3}, Z_{3}^{\prime} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$ computing $I D_{n+1}=H\left(C_{0}, C_{2}\right) \in \mathbb{Z}_{N}$ and testing whether the following equation holds:

$$
\begin{equation*}
e\left(g \cdot Z_{3}, C_{1}\right) \stackrel{?}{=} e\left(C_{0},\left(h \cdot u_{n+1}^{I D_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_{i}^{I D_{i}} \cdot Z_{3}^{\prime}\right)\right) \tag{1}
\end{equation*}
$$

If so, the decryption algorithm simply invokes $\operatorname{Decrypt}_{\mathrm{CPA}}\left(\mathbf{V} \cup\left\{\left(I D_{n+1}\right)\right\}, C T, S K_{\mathbf{I D}}\right)$ to get message $M$. Otherwise, the ciphertext is invalid and the decryption algorithm simply outputs $N U L L$.

Remark 1. Note that the above ciphertext validity test can be done publicly since it only involves public parameters and ciphertexts. It is useful for our scheme to build advanced protocols, e.g., publicly verifiable HIBBE encryption with CCA2 security [13, 26, 38]. Also, it allows a gateway or firewall to filter spams (i.e., invalid ciphertexts) without requiring the secret keys of the receivers. Similar functionality has been applied to identify dishonest transactions in mobile E-commerce scenario [24].

Soundness. If the ciphertext is legitimate, then the following tuple

$$
\left(g, C_{0}=g^{\beta},\left(h \cdot u_{n+1}^{I D_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_{i}^{I D_{i}}\right), C_{2}=\left(h \cdot u_{n+1}^{I D_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_{i}^{I D_{i}}\right)^{\beta}\right)
$$

is a valid Diffie-Hellman tuple. Elements $Z_{3}, Z_{3}^{\prime} \in \mathbb{G}_{p_{3}}$ can be eliminated in both sides of Equation (1) with the orthogonality property. Accordingly, Equation (1) holds. Also, this ciphertext is a valid $\operatorname{HIBBE}_{\mathrm{CPA}}$ ciphertext for the receiver identity vector set $\mathbf{V} \cup\left\{\left(I D_{n+1}\right)\right\}$ with $I D_{n+1}=H\left(C_{0}, C_{2}\right)$. Since $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V}) \subseteq \mathbf{V} \cup\left\{\left(I D_{n+1}\right)\right\}$, the decryption algorithm can decrypt the ciphertext by invoking the underlying $\mathbf{D e c r y p t}_{\mathrm{CPA}}\left(\mathbf{V} \cup\left\{\left(I D_{n+1}\right)\right\}, C T, S K_{\mathbf{I D}}\right)$.

### 5.3 Security Analysis

We now allow decryption queries in all games defined previously in Section 4.2. Our simulation works as follows. When receiving a decryption query from the adversary, the simulator first checks Equation (1) to determine whether the ciphertext is valid. If the equality holds, the simulator generates a secret key for any identity vector ID satisfying that $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$, and then uses this key to decrypt the ciphertext. In the challenge phase, the simulator creates a challenge ciphertext $C T^{*}=\left(C_{0}^{*}, C_{1}^{*}, C_{2}^{*}\right)$ for the challenge identity vector set $\mathbf{V}^{*} \cup\left\{\left(I D_{n+1}^{*}\right)\right\}$, where $I D_{n+1}^{*}=H\left(C_{0}^{*}, C_{2}^{*}\right)$. Since the hash function $H$ is collision resistant, the adversary is unable to make any valid ciphertext queries that would require the simulator to use a identity vector set $\mathbf{V} \cup\left\{\left(I D_{n+1}^{\prime}\right)\right\}$ with $I D_{n+1}^{\prime}=I D_{n+1}^{*}$. Note that the adversary cannot issue secret key query for the dummy user because it is not available before the simulator produces the challenge ciphertext. Hence, the simulation can be done by invoking the underlying $\operatorname{HIBBE}_{\text {CPA }}$.

Formally, the CCA2 security of the above scheme is guaranteed by the following Theorem.
Theorem 2. Let $\mathbb{G}$ be a group (of composite order $N$ ) endowed with an efficient bilinear map. Suppose all the three assumptions defined in Definition 1, Definition 2 and Definition 3 hold in $\mathbb{G}$. Then our $H I B B E_{C C A 2}$ is IND-CIVS-CCA2 secure.

Similarly to those in CPA security proofs, we denote those games respectively by GameCCA2 $\mathbf{R e a l}$, GameCCA2 Restricted , GameCCA2 $\mathbf{k}^{\text {w }}$ with $k \in[0, q]$ and GameCCA2 $\mathbf{F i n a l}$. For a security parameter $\lambda$, we respectively represent the advantages of winning in these games by $\operatorname{Adv}_{\text {Real }}^{\mathrm{CCA} 2}(\lambda)$, $A d v_{\text {Restricted }}^{\mathrm{CCA} 2}(\lambda), \operatorname{Adv} v_{k}^{\mathrm{CCA} 2}(\lambda)$ with $k \in[0, q]$, and $\operatorname{Adv} v_{\text {Final }}^{\mathrm{CCA} 2}(\lambda)$. The security of our HIBBE $\mathrm{CCA}_{2}$ follows from the indistinguishability between the these games, assuming that the three assumptions defined in Section 2 hold.

Lemma 5. Suppose that Assumption 2 holds. Then there is no polynomial time algorithm that can distinguish GameCCA2 Real $_{\text {Real }}$ from GameCCA $\boldsymbol{\mathcal { L }}_{\text {Restricted }}$ with non-negligible advantage.

Proof. The proof of this lemma is identical with the proof of lemma 1.
Lemma 6. There is no polynomial time algorithm that can distinguish GameCCA2 $\boldsymbol{\mathcal { L }}_{\text {Restricted }}$ from GameCCA2 ${ }_{0}$ with non-negligible advantage assuming that Assumption 1 holds.

Proof. Assume that there exists an adversary $\mathcal{A}$ that can distinguish GameCCA2 Restricted from GameCCA2 $2_{0}$ with advantage $\epsilon_{0}$. We build an algorithm $\mathcal{B}$ that can refute Assumption 1 with advantage $\operatorname{Adv} 1_{\mathcal{B}}(\lambda) \geq \epsilon_{0} . \mathcal{B}$ takes the challenge tuple $\left(g, X_{3}, T\right)$ as inputs. The goal of $\mathcal{B}$ is to determine whether $T$ is an element in $\mathbb{G}_{p_{1}}$ or an element in $\mathbb{G}_{p_{1} p_{2}}$. In the Setup phase, $\mathcal{B}$ randomly chooses exponents $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and $\gamma_{i} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $i \in[0, n+1]$. It sets $h \leftarrow g^{\gamma_{0}}$ and $u_{i} \leftarrow g^{\gamma_{i}}$ for all $i \in[1, n+1]$. Finally, $\mathcal{B}$ gives the public key

$$
P K \leftarrow\left(g, h, u_{1}, \cdots, u_{n}, u_{n+1}, X_{3}, e(g, g)^{\alpha}\right)
$$

to $\mathcal{A}$. Note that $\mathcal{B}$ knows the master key $M S K \leftarrow g^{\alpha}$.
For a secret key query with identity vector $\mathbf{I D}=\left(I D_{1}, \cdots, I D_{d}\right)$ issued by $\mathcal{A}, \mathcal{B}$ runs the usual key generation algorithm to return the secret key.

When receiving a decryption query from $\mathcal{A}$ with a ciphertext $C T=\left(C_{0}, C_{1}, C_{2}\right)$ and a receiver identity vector set $\mathbf{V}, \mathcal{B}$ first computes $I D_{n+1}=H\left(C_{0}, C_{2}\right)$ and determines whether the ciphertext is valid by checking Equation (1) defined in Section 5.2. If the equality does not hold, then the ciphertext is invalid and $\mathcal{B}$ returns $N U L L$. Otherwise, $\mathcal{B}$ generates a normal key for any user whose identity vector is $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$ using the master key $g^{\alpha}$. Then, $\mathcal{B}$ uses this key to decrypt the ciphertext and returns the extracted message to $\mathcal{A}$.

In the challenge phase, $\mathcal{A}$ outputs two equal-length messages $M_{0}, M_{1} \in \mathbb{G}_{T}$, together with a challenge identity vector set $\mathbf{V}^{*}$. Denote $\mathbb{I}^{*}=\left\{i: I D_{i}^{*} \in S_{\mathbf{V}^{*}}\right\}$. $\mathcal{B}$ flips a random coin $b \stackrel{R}{\leftarrow}\{0,1\}$ and returns the challenge ciphertext

$$
C T^{*} \leftarrow\left(C_{0}^{*}, C_{1}^{*}, C_{2}^{*}\right) \leftarrow\left(T, T^{\gamma_{0}+\sum_{i \in \mathbb{R}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}}, M_{b} \cdot e\left(g^{\alpha}, T\right)\right)
$$

where $I D_{n+1}^{*}=H\left(C_{0}^{*}, C_{2}^{*}\right)=H\left(T, M_{b} \cdot e\left(g^{\alpha}, T\right)\right)$.
Note that the components in the challenge ciphertext do not involve elements in $\mathbb{G}_{p_{3}}$. Therefore, for any randomly chosen elements $Z_{3}, Z_{3}^{\prime} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$, the challenge ciphertext is valid due to the following equalities:

$$
\frac{e\left(g \cdot Z_{3}, C_{1}^{*}\right)}{e\left(C_{0}^{*},\left(h \cdot u_{n+1}^{I D_{n+1}^{*}} \cdot \prod_{i \in \mathbb{I}^{*}} u_{i}^{I D_{i}^{*}}\right) \cdot Z_{3}^{\prime}\right)}=\frac{e\left(g \cdot Z_{3}, T^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}}\right)}{e\left(T, g^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}} \cdot Z_{3}^{\prime}\right)}=1
$$

Finally, $\mathcal{A}$ outputs a bit $b$ as its guess of it is in GameCCA2 Restricted or in GameCCA2 $\boldsymbol{N}_{0}$. If $\mathcal{A}$ guesses that $\mathcal{A}$ is in GameCCA2 Restricted , $\mathcal{B}$ outputs $T \in \mathbb{G}_{p_{1}}$. Otherwise, $\mathcal{B}$ concludes $T \in \mathbb{G}_{p_{1} p_{2}}$.

The decryption query can be responded to perfectly, since $\mathcal{B}$ can generate normal keys for arbitrary identity vectors using the master key $g^{\alpha}$. With the identical analysis showed in the proof of Lemma 1 , if $\mathcal{A}$ has advantage $\epsilon_{0}$ in distinguishing GameCCA2 $2_{\text {Restricted }}$ and GameCCA2 $_{0}$, then $\mathcal{B}$ can determine the distribution of $T$ with advantage $\operatorname{Adv} 1_{\mathcal{B}}(\lambda) \geq \epsilon_{0}$.

Lemma 7. If Assumption 2 holds, then no polynomial time algorithm can distinguish $\boldsymbol{G a m e} \boldsymbol{C C A} \boldsymbol{\mathcal { L }}_{k-1}$ from $\boldsymbol{G a m e} \boldsymbol{C C A} \boldsymbol{2}_{k}$ with non-negligible advantage.

Proof. Assume there is an adversary $\mathcal{A}$ that can distinguish GameCCA2 ${ }_{\mathrm{k}-1}$ from GameCCA2 $\mathbf{k}_{\mathrm{k}}$ with advantage $\epsilon_{k}$. Then, by invoking $\mathcal{A}$ as a blackbox, we can construct an algorithm $\mathcal{B}$ that refutes Assumption 2 with advantage $A d v 2_{\mathcal{B}}(\lambda) \geq \epsilon_{k}$. The input of $\mathcal{B}$ is an instance $\left(g, X_{1} X_{2}, X_{3}, Y_{2} Y_{3}, T\right)$ from the second assumption. $\mathcal{B}$ has to decide whether $T$ is an element in $\mathbb{G}_{N}$ or an element in $\mathbb{G}_{p_{1} p_{3}}$. $\mathcal{B}$ randomly chooses $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ and $\gamma_{i} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $i \in[1, n+1]$. It sends $\mathcal{A}$ the public key

$$
P K \leftarrow\left(g, h, u_{1}, \cdots, u_{n}, u_{n+1}, X_{3}, e(g, g)^{\alpha}\right)
$$

with $h \leftarrow g^{\gamma_{0}}$ and $u_{i} \leftarrow g^{\gamma_{i}}$ for all $i \in[1, n+1]$. The master key is $M S K \leftarrow g^{\alpha}$ and is kept by $\mathcal{B}$.
When receiving the secret key query with an identity vector $\mathbf{I D}=\left(I D_{1}, \cdots, I D_{d}\right), \mathcal{B}$ runs the same as Phase 1 in Lemma 3 to generate the secret key and returns it to $\mathcal{A}$.

When $\mathcal{A}$ issues a decryption query for a ciphertext $C T=\left(C_{0}, C_{1}, C_{2}\right)$ with a receiver identity vector set $\mathbf{V}, \mathcal{B}$ sets $I D_{n+1}=H\left(C_{0}, C_{2}\right)$ and checks Equation (1) described in Section 5.2. If the equality holds, $\mathcal{B}$ creates a normal key for any identity vector $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$ and returns the message decrypted from the ciphertext $C T$. Otherwise it returns $N U L L$ since the ciphertext is invalid.

In the Challenge phase, $\mathcal{A}$ outputs two equal-length messages $M_{0}, M_{1} \in \mathbb{G}_{T}$, together with an identity vector set $\mathbf{V}^{*}$ as the challenge identity vector set. Denote $\mathbb{I}^{*}=\left\{i: I D_{i}^{*} \in S_{\mathbf{V}^{*}}\right\} . \mathcal{B}$ chooses a random bit $b \stackrel{R}{\leftarrow}\{0,1\}$ and outputs the resulting ciphertext

$$
C T^{*} \leftarrow\left(C_{0}^{*}, C_{1}^{*}, C_{2}^{*}\right) \leftarrow\left(X_{1} X_{2},\left(X_{1} X_{2}\right)^{\gamma_{0}+\sum_{i \in \mathbb{1}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}}, M_{b} \cdot e\left(g, X_{1} X_{2}\right)^{\alpha}\right)
$$

where $I D_{n+1}^{*}=H\left(C_{0}^{*}, C_{2}^{*}\right)=H\left(X_{1} X_{2}, e\left(g, X_{1} X_{2}\right)^{\alpha}\right)$. Equation (1) holds for this ciphertext since for any $Z_{3}, Z_{3}^{\prime} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$,

$$
\frac{e\left(g \cdot Z_{3}, C_{1}^{*}\right)}{e\left(C_{0}^{*},\left(h \cdot u_{n+1}^{I D_{n+1}^{*}} \cdot \prod_{i \in \mathbb{I}^{*}} u_{i}^{I D_{i}^{*}}\right) \cdot Z_{3}^{\prime}\right)}=\frac{e\left(g \cdot Z_{3},\left(X_{1} X_{2}\right)^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}}\right)}{e\left(X_{1} X_{2}, g^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}} \cdot Z_{3}^{\prime}\right)}=1
$$

Therefore, this ciphertext is valid.
Note that this ciphertext is semi-functional by implicitly setting

$$
y_{c}=\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}
$$

Since from GameCCA2 Restricted , $\mathcal{A}$ cannot issue a secret key query with the identity vector that is a prefix of the challenge receiver identity vector set module $p_{2}, y_{c}$ and $y_{k}$ will seem randomly
distribute to $\mathcal{A}$. Therefore, the relationship between $y_{c}$ and $y_{k}$ does not give any advantage to $\mathcal{A}$ for distinguishing between the two games.

Though the relationship between $y_{c}$ and $y_{k}$ is hidden from $\mathcal{A}$, this special setting disallows $\mathcal{B}$ itself to test whether the $k^{\text {th }}$ key for identity vector ID is semi-functional. The method is to generate a semi-functional ciphertext for any identity vector set $\mathbf{V}$ such that $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$ and to decrypt it using the $k^{\text {th }}$ key. If the $k^{\text {th }}$ key is normal, the decryption is correct. However, if the $k^{\text {th }}$ key is semi-functional, then by the definition of semi-functional secret key, the $k^{\text {th }}$ key cannot decrypt the semi-functional ciphertext. In this way, $\mathcal{B}$ may have advantage 1 to answer $T \in \mathbb{G}_{N}$ or $T \in \mathbb{G}_{p_{1} p_{2} p_{3}}$ without $\mathcal{A}$ 's help.

In fact, this well-designed secret key generated in the $k^{\text {th }}$ key query disallows $\mathcal{B}$ to use this method. If $\mathcal{B}$ tries to do that, then no matter whether the $k^{\text {th }}$ key is normal or semi-functional, decryption will always work, because $y_{k}+\sum_{i \in \mathbb{I} \backslash \mathrm{I}} I D_{i} \cdot \gamma_{i}+I D_{n+1} \cdot \gamma_{n+1}=y_{c}$, where $\mathrm{I}=\left\{i: I D_{i} \in S_{\mathbf{I D}}\right\}$ and $\mathbb{I}=\left\{i: I D_{i} \in S_{\mathbf{V}}\right\}$. In other words, for the $k^{\text {th }}$ secret key query, $\mathcal{B}$ can only generate a nominally semi-functional key. Hence decryption is always correct by the definition of nominally semi-functional key given in Section 4.2.

If $\mathcal{A}$ outputs the guess that it is in GameCCA2 $\mathbf{k}_{\mathbf{k}-\mathbf{1}}, \mathcal{B}$ answers $T \in \mathbb{G}_{p_{1} p_{3}}$. Otherwise, $\mathcal{A}$ outputs that it is in $\mathbf{G a m e C C A}_{\mathbf{k}}$, and $\mathcal{B}$ decides $T \in \mathbb{G}_{N}$.

With the similar reason in the proof of Lemma 3, if $\mathcal{A}$ has advantage $\epsilon_{k}$ in distinguishing the game GameCCA2 $_{\mathbf{k}-1}$ from the game $\mathbf{G a m e C C A}_{\mathbf{k}}, \mathcal{B}$ can distinguish $T \in \mathbb{G}_{p_{1} p_{3}}$ from $T \in \mathbb{G}_{N}$ with advantage $\operatorname{Adv} 2_{\mathcal{B}}(\lambda) \geq \epsilon_{k}$.

Lemma 8. Suppose that Assumption 3 holds. No polynomial time algorithm that can distinguish GameCCA2 $\mathcal{L}_{q}$ from GameCCA1 $\mathcal{L}_{\text {Final }}$ with non-negligible advantage.

Proof. Assume $\mathcal{A}$ can distinguish GameCCA2 $\mathbf{q}_{\mathbf{q}}$ from GameCCA2 Final with advantage $\epsilon_{F}$. By invoking $\mathcal{A}$ as a blackbox, we build an algorithm $\mathcal{B}$ refuting Assumption 3 with advantage $A d v 3_{\mathcal{B}}(\lambda) \geq$ $\epsilon_{F}$. The input of $\mathcal{B}$ is the challenge tuple $\left(g, g^{\alpha} X_{2}, X_{3}, g^{s} Y_{2}, Z_{2}, T\right)$ of Assumption 3. $\mathcal{B}$ has to answer whether $T$ is $e(g, g)^{\alpha s}$ or a random element in $\mathbb{G}_{T}$. $\mathcal{B}$ randomly chooses $\gamma_{i} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $i \in[0, n+1]$ and sets the public key

$$
P K \leftarrow\left(g=g, h=g^{\gamma_{0}}, u_{1}=g^{\gamma_{1}}, \cdots, u_{n}=g^{\gamma_{n}}, u_{n+1}=g^{\gamma_{n+1}}, X_{3}, e(g, g)^{\alpha}=e\left(g^{\alpha} X_{2}, g\right)\right)
$$

When $\mathcal{A}$ requests a secret key for an identity vector $\mathbf{I D}, \mathcal{B}$ chooses random exponents $w_{0}, w_{1}, t_{0}, t_{1} \stackrel{R}{\leftarrow}$ $\mathbb{Z}_{N}$ and $v_{j}, z_{j} \stackrel{R}{\leftarrow} \mathbb{Z}_{N}$ for all $j \in[1, n] \backslash \mathrm{I}$, where $\mathrm{I}=\left\{i: I D_{i} \in S_{\mathrm{ID}}\right\}$. Then, $\mathcal{B}$ outputs the secret key

$$
S K_{\mathrm{ID}} \leftarrow\left(g^{\alpha} X_{2}\left(h \cdot \prod_{i \in \mathrm{I}} u_{i}^{I D_{i}}\right)^{r} Z_{2}^{t_{0}} X_{3}^{w_{0}}, g^{r} Z_{2}^{t_{1}} X_{3}^{w_{1}},\left\{u_{j}^{r} Z_{2}^{z_{j}} X_{3}^{v_{j}}\right\}_{j \in[1, n] \backslash \mathrm{I}}\right)
$$

Note that the resulting key is semi-functional.
When $\mathcal{B}$ receives a decryption query for a ciphertext $C T=\left(C_{0}, C_{1}, C_{2}\right)$ associated with a receiver identity vector set $\mathbf{V}$, it first sets $I D_{n+1}=H\left(C_{0}, C_{2}\right)$. Then, $\mathcal{B}$ checks Equation (1) to verify the validity of $C T$. If the equality does not hold, $\mathcal{B}$ simply returns $N U L L$. Otherwise, since $\mathcal{B}$ knows a random generator $g$ of $\mathbb{G}_{p_{1}}$ and a random element $X_{3} \in \mathbb{G}_{p_{3}}$, it can run the same algorithm described in Phase 1 to generate a semi-functional secret key for $\mathbf{I D} \in \operatorname{Pref}(\mathbf{V})$ and use it to decrypt $C T$.

Although the generated secret keys are all semi-functional, $\mathcal{B}$ can use them to correctly respond the decryption queries. The reason is that $\mathcal{A}$ can only issue valid normal ciphertexts for decryption queries. One one hand, $\mathcal{A}$ cannot generate semi-functional ciphertexts for any identity vector sets $\mathbf{V}$ without the knowledge of the subgroup $\mathbb{G}_{p_{2}}$, except for the challenge identity vector set. Otherwise $\mathcal{A}$ can distinguish the preceding security games by issuing a secret key query for an identity vector ID $\in \operatorname{Pref}(\mathbf{V})$ and try to decrypt by itself. This has been prevented in the CPA security proof. On the other hand, only semi-functional ciphertexts that can be obtained by $\mathcal{A}$ are the ones modified from the challenge ciphertext. However, any modifications done by $\mathcal{A}$ without the knowledge of the subgroup $\mathbb{G}_{p_{2}}$ for the challenge ciphertext can be detected by Equation (1). Therefore, any decryption queries for semi-functional ciphertexts would be prevented. The secret keys would only be used to decrypt normal ciphertexts and the decryption queries can be responded correctly.

When suitable, $\mathcal{A}$ outputs two equal-length messages $M_{0}, M_{1} \in \mathbb{G}_{T}$, and a challenge identity vector set $\mathbf{V}^{*}$. Denote $\mathbb{I}^{*}=\left\{i: I D_{i} \in S_{\mathbf{V}^{*}}\right\}$. $\mathcal{B}$ chooses a random bit $b \stackrel{R}{\leftarrow}\{0,1\}$ and outputs

$$
C T^{*} \leftarrow\left(C_{0}^{*}, C_{1}^{*}, C_{2}^{*}\right) \leftarrow\left(g^{s} Y_{2},\left(g^{s} Y_{2}\right)^{\gamma_{0}+\sum_{i \in \mathbb{I}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}}, M_{b} \cdot T\right)
$$

where $I D_{n+1}^{*}=H\left(C_{0}^{*}, C_{2}^{*}\right)=H\left(g^{s} Y_{2}, M_{b} \cdot T\right)$. Note that for any $Z_{3}, Z_{3}^{\prime} \stackrel{R}{\leftarrow} \mathbb{G}_{p_{3}}$,

$$
\frac{e\left(g \cdot Z_{3}, C_{1}^{*}\right)}{e\left(C_{0}^{*},\left(h \cdot u_{n+1}^{I D_{n+1}^{*}} \cdot \prod_{i \in \mathbb{I}^{*}} u_{i}^{I D_{i}^{*}} \cdot Z_{3}^{\prime}\right)\right)}=\frac{e\left(g \cdot Z_{3},\left(g^{s} Y_{2}\right)^{\gamma_{0}+\sum_{i \in \mathbb{1}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}}\right)}{e\left(g^{s} Y_{2}, g^{\gamma_{0}+\sum_{i \in \mathbb{1}^{*}} I D_{i}^{*} \cdot \gamma_{i}+I D_{n+1}^{*} \cdot \gamma_{n+1}} \cdot Z_{3}^{\prime}\right)}=1
$$

Hence $C T^{*}$ is a valid ciphertext.
$\mathcal{B}$ answers $T \leftarrow e(g, g)^{\alpha s}$ if $\mathcal{A}$ outputs the guess that it is in GameCCA2 $\mathbf{q}_{\mathbf{q}}$. Otherwise, $\mathcal{B}$ determines $T \stackrel{R}{\leftarrow} \mathbb{G}_{T}$ if $\mathcal{A}$ guesses that it is in GameCCA2 Final .

Similar to the analysis of Lemma $4, \mathcal{B}$ can distinguish $T \leftarrow e(g, g)^{\alpha s}$ from a random element in $\mathbb{G}_{T}$ with advantage $\operatorname{Adv} 3_{\mathcal{B}}(\lambda) \geq \epsilon_{F}$ if $\mathcal{A}$ has advantage $\epsilon_{F}$ in distinguishing GameCCA2 $\mathbf{q}_{\mathbf{q}}$ from GameCCA2 ${ }_{\text {Final }}$.

With the four lemmas described above, the security proof of Theorem 2 follows.
Proof. Since in GameCCA2 Final , the ciphertext has been replaced with a random element in $\mathbb{G}_{T}$, the value of $b$ chosen by the challenger is information-theoretically hidden from $\mathcal{A}$. Hence $\mathcal{A}$ can obtain no advantage in breaking our HIBBES. By combining the four lemmas shown previously, we have that

$$
\begin{aligned}
\left|A d v_{\text {Real }}^{\mathrm{CCA} 2}(\lambda)\right| & \leq \mid A d v_{\text {Real }}^{\mathrm{CCA} 2}(\lambda)-A d v_{\text {Restricted }}^{\mathrm{CCA} 2}(\lambda)+A d v_{\text {Restricted }}^{\mathrm{CCA} 2}(\lambda)-\cdots-\operatorname{Adv_{\text {Final}}^{\mathrm {CCA}2}(\lambda )+Adv_{\text {FCinal}}^{\mathrm {CCA}2}(\lambda )|} \\
& \leq\left|A d v_{\text {Real }}^{\mathrm{CCA} 2}(\lambda)-\operatorname{Adv} v_{\text {Restricted }}^{\mathrm{CCA}}(\lambda)\right|+\cdots+\left|A d v_{q}^{\mathrm{CCA} 2}(\lambda)-\operatorname{Adv} v_{\text {Final }}^{\mathrm{CA} 2}(\lambda)\right|+\left|A d v_{\text {Final }}^{\mathrm{CCA} 2}(\lambda)\right| \\
& \leq 2 \cdot \operatorname{Adv2_{\mathcal {A}}(\lambda )+\operatorname {Adv}1_{\mathcal {A}}(\lambda )+q\cdot Adv2_{\mathcal {A}}(\lambda )+Adv3_{\mathcal {A}}(\lambda )}
\end{aligned}
$$

If the three assumptions hold, then for all polynomial time $\mathcal{A}, \operatorname{Adv} 1_{\mathcal{A}}(\lambda), \operatorname{Adv} 2_{\mathcal{A}}(\lambda)$, and $\operatorname{Adv} 3_{\mathcal{A}}(\lambda)$ are all negligible probability. Hence for all polynomial time algorithms, the advantage of breaking our $\mathrm{HIBBE}_{\mathrm{CCA} 2}$ is negligible.

### 5.4 Efficient Tradeoff Between Ciphertext Size and Key Size

The public/secret key size and ciphertext size in $(D, n)$ - HIBBE $_{\text {CCA2 }}$ remain the same as those of the underlying $(D, n+1)-\mathrm{HIBBE}_{\text {CPA }}$ system. The encryption algorithm needs only one more hash operation. The decryption algorithm does one more hash operation and one more extra test of Equation (1) in which a two-base pairing is required and $u_{i}^{I D_{i}}$ can be pre-computed for $i \in[1, n]$. Table 2 shows comparisons between our CPA-secure ( $D, n+1$ )-HIBBE and our CCA2-secure ( $D, n$ )-HIBBE in detail. In Table 2, the secret key $S K_{\text {ID }}$ is associated with the identity vector ID, and the ciphertext $C T$ is associated with the receiver identity vector set $\mathbf{V}$. We denote $\tau_{e}$ as one exponent operation time in $\mathbb{G}, \tau_{m}$ as one multiplication operation time in $\mathbb{G}, \tau_{p}$ as one pairing operation time in $\mathbb{G}$, and $\tau_{h}$ as one hash operation time for the hash function $H$. From Table 2, it can be seen that the additional overheads are marginal.

HIBBE with Shorter Secret Keys. In our HIBBES, while the ciphertext contains only three group elements, the secret key for user at depth $d$ contains $n-d+2$ elements. In some scenarios, e.g., when the storage capacities of the receivers are limited, one may expect an efficient tradeoff between key size and ciphertext size. Note that users in an HIBBES are organized as a tree $T$ with $n$ nodes (PKG as the sink is not countered). We divide $T$ into $\mathcal{T}$ subtrees with $n_{i}$ nodes, where $i \in[1, \mathcal{T}]$. To achieve better balance, as shown in Figure 3, all the subtrees may be obtained in a way satisfying:

1. The number of nodes for each subtree is approximately equal. That is, for the $i^{\text {th }}$ subtree with $i \in[1, \mathcal{T}]$, we have $n_{i} \approx n / \mathcal{T}$;
2. If possible, all subtrees share minimum number of higher-level nodes.

Table 2. Comparison Between CPA-secure ( $D, n+1$ )-HIBBE and CCA2-secure ( $D, n$ )-HIBBE

|  | $(D, n+1)-$ HIBBE $_{\text {CPA }}$ | $(D, n)-\mathrm{HIBBE}_{\mathrm{CCA} 2}$ |
| :--- | :---: | :---: |
| Active Users | $n+1$ | $n$ |
| $P K$ Size | $n+5$ | $n+5$ |
| $S K_{\text {ID Size }}$ | $n-\\|\mathbf{I D}\\|+2$ | $n-\\|\mathbf{I D}\\|+2$ |
| $C T$ Size | 3 | 3 |
| Encryption Time | $\left(2+\left\|S_{\mathbf{V}}\right\|\right) \cdot\left(\tau_{e}+\tau_{m}\right)$ | $\left(2+\left\|S_{\mathbf{V}}\right\|\right) \cdot\left(\tau_{e}+\tau_{m}\right)+\tau_{h}$ |
| Decryption Time | $\leq\left(1+\left\|S_{\mathbf{V}}\right\|\right) \cdot\left(\tau_{e}+\tau_{m}\right)+2 \tau_{p} \mid \leq\left(1+\left\|S_{\mathbf{v}}\right\|\right) \cdot\left(\tau_{e}+\tau_{m}\right)+4 \tau_{p}+\tau_{h}$ |  |

We then implement independent HIBBE instances in each subtree. When broadcasting, one encrypts the messages with each instance where the broadcast subsets are the intersection of the original broadcast set and the subtrees. Each receiver can decrypt the ciphertext component corresponding to its subtree. It is clear that, by using this subtree method, the key size is $O\left(\frac{n}{\mathcal{T}}\right)$ and the ciphertext size is $O(\mathcal{T})$. By setting $\mathcal{T}=\sqrt{n}$, both the key size and the ciphertext size are $O(\sqrt{n})$.


Fig. 2. Constant Size Ciphertext HIBBE.


Fig. 3. Shorter Secret keys HIBBE.

## 6 Conclusion

This paper extended the functionality of HIBE to HIBBE, allowing users to encrypt to multiple receivers organized in hierarchy, while supporting delegation of secret keys to relieve the private key generator from heavy key management burden. The new cryptographic primitive offers a novel avenue to establish secure data sharing systems, or suitable distributed computation and communication applications. We constructed a CPA-secure HIBBES with short ciphertexts. We then proposed a transformation technique to convert our basic scheme to obtain CCA2-security. Both schemes are efficient and proven to be fully secure under three static assumptions in the standard model.

## Acknowledgment

This paper is partially supported by the National Key Basic Research Program (973 program) through project 2012CB315905, by the Natural Science Foundation of China through projects 61370190 ,
$61173154,61272501,61402029,61472429,61202465$ and 61003214 , by the Beijing Natural Science Foundation through project 4132056, by the Guangxi natural science foundation through project 2013GXNSFBB053005, by the Fundamental Research Funds for the Central Universities, the Research Funds (No. 14XNLF02) of Renmin University of China, the Innovation Fund of China Aerospace Science and Technology Corporation, Satellite Application Research Institute through project 2014-CXJJ-TX-10, the Open Project of Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University and the Open Research Fund of Beijing Key Laboratory of Trusted Computing.

## References

1. Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: EUROCRYPT 2004. Volume 3027 of LNCS., Springer Berlin Heidelberg (2004) 223-238
2. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: EUROCRYPT 2005. Volume 3494 of LNCS., Springer Berlin Heidelberg (2005) 440-456
3. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: CRYPTO 2001. Volume 2139 of LNCS., Springer Berlin Heidelberg (2001) 213-229
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. SIAM Journal on Computing $32(3)$ (2003) 586-615
5. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: CRYPTO 2005. Volume 3621 of LNCS., Springer Berlin Heidelberg (2005) 258-275
6. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: TCC 2005. Volume 3378 of LNCS., Springer Berlin Heidelberg (2005) 325-341
7. Boneh, D., Hamburg, M.: Generalized identity based and broadcast encryption schemes. In: ASIACRYPT 2008. Volume 5350 of LNCS., Springer Berlin Heidelberg (2008) 455-470
8. Boneh, D., Katz, J.: Improved efficiency for cca-secure cryptosystems built using identity-based encryption. In: CT-RSA 2005. Volume 3376 of LNCS., Springer Berlin Heidelberg (2005) 87-103
9. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: CCS 2005, ACM (2005) 320-329
10. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: EUROCRYPT 2003. Volume 2656 of LNCS., Springer Berlin Heidelberg (2003) 255-271
11. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: EUROCRYPT 2004. Volume 3027 of LNCS., Springer Berlin Heidelberg (2004) 207-222
12. Chen, H.C.: A trusted user-to-role and role-to-key access control scheme. Soft Computing 20(5) (2016) 1721-1733
13. Chen, X., Li, J., Huang, X., Ma, J., Lou, W.: New publicly verifiable databases with efficient updates. IEEE Transactions on Dependable and Secure Computing $12(5)$ (2015) 546-556
14. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: ASIACRYPT 2007. Volume 4833 of LNCS., Springer Berlin Heidelberg (2007) 200-215
15. Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: Pairing 2007. Volume 4575 of LNCS., Springer Berlin Heidelberg (2007) 39-59
16. Deng, H., Wu, Q., Qin, B., Domingo-Ferrer, J., Zhang, L., Liu, J., Shi, W.: Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts. Information Sciences 275 (2014) 370-384
17. Dodis, Y., Fazio, N.: Public key broadcast encryption for stateless receivers. In: Digital Rights Management. Volume 2696 of LNCS., Springer Berlin Heidelberg (2003) 61-80
18. Fiat, A., Naor, M.: Broadcast encryption. In: CRYPTO 1993. Volume 773 of LNCS., Springer Berlin Heidelberg (1994) 480-491
19. Gentry, C., Halevi, S.: Hierarchical identity based encryption with polynomially many levels. In: TCC 2009. Volume 5444 of LNCS., Springer Berlin Heidelberg (2009) 437-456
20. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. In: ASIACRYPT 2002. Volume 2501 of LNCS., Springer Berlin Heidelberg (2002) 548-566
21. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: EUROCRYPT 2009. Volume 5479 of LNCS., Springer Berlin Heidelberg (2009) 171-188
22. Halevy, D., Shamir, A.: The lsd broadcast encryption scheme. In: CRYPTO 2002. Volume 2442 of LNCS., Springer Berlin Heidelberg (2002) 47-60
23. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: EUROCRYPT 2002. Volume 2332 of LNCS., Springer Berlin Heidelberg (2002) 466-481
24. Huan, J., Yang, Y., Huang, X., Yuen, T.H., Li, J., Cao, J.: Accountable mobile e-commerce scheme via identity-based plaintext-checkable encryption. Information Sciences 345 (2016) 143-155
25. Huang, X., Liu, J.K., Hua, S., Xiang, Y., Liang, K., Zhou, J.: Cost-effective authentic and anonymous data sharing with forward security. IEEE Transactions on Computers 64(4) (2015) 971-983
26. Huang, X., Xiang, Y., Chonka, A., Zhou, J., Deng, R.H.: A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. IEEE Transactions on Parallel and Distributed Systems 22(8) (2011) 1390-1397
27. Kim, J., Susilo, W., Au, M.H., Seberry, J.: Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext. IEEE Transactions on Information Forensics and Security 10(3) (2015) 679-693
28. Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure hibe with short ciphertexts. In: TCC 2010. Volume 5978 of LNCS., Springer Berlin Heidelberg (2010) 455-479
29. Lewko, A., Waters, B.: New proof methods for attribute-based encryption: Achieving full security through selective techniques. In: CRYPTO 2012. Volume 7417 of LNCS., Springer Berlin Heidelberg (2012) 180-198
30. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In: PKC 2012. Volume 7293 of LNCS., Springer Berlin Heidelberg (2012) 206-224
31. Liu, W., Liu, J., Wu, Q., Qin, B.: Hierarchical identity-based broadcast encryption. In: ACISP 2014. Volume 8544 of LNCS., Springer Berlin Heidelberg (2014) 242-257
32. Liu, W., Liu, J., Wu, Q., Qin, B., Li, Y.: Practical chosen-ciphertext secure hierarchical identity-based broadcast encryption. International Journal of Information Security 15(1) (2016) 35-50
33. Liu, Z., Weng, J., Li, J., Yang, J., Fu, C., Jia, C.: Cloud-based electronic health record system supporting fuzzy keyword search. Soft Computing (2015)
34. Park, J.H., Lee, K., Lee, D.H.: New chosen-ciphertext secure identity-based encryption with tight security reduction to the bilinear diffie-hellman problem. Information Sciences 325 (2015) 256-270
35. Qin, B., Wu, Q., Zhang, L., Farràs, O., Domingo-Ferrer, J.: Provably secure threshold public-key encryption with adaptive security and short ciphertexts. Information Sciences 210 (2012) 67-80
36. Ren, Y., Gu, D.: Fully cca2 secure identity based broadcast encryption without random oracles. Information Processing Letters 109(11) (2009) 527-533
37. Shamir, A.: Identity-based cryptosystems and signature schemes. In: CRYPTO 1984. Volume 196 of LNCS., Springer Berlin Heidelberg (1985) 47-53
38. Wang, J., Chen, X., Huang, X., You, I., Xiang, Y.: Verifiable auditing for outsourced database in cloud computing. IEEE Transactions on Computers 64(11) (2015) 3293-3303
39. Waters, B.: Efficient identity-based encryption without random oracles. In: EUROCRYPT 2005. Volume 3494 of LNCS., Springer Berlin Heidelberg (2005) 114-127
40. Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In: CRYPTO 2009. Volume 5677 of LNCS., Springer Berlin Heidelberg (2009) 619-636
41. Wu, Q., Qin, B., Zhang, L., Domingo-Ferrer, J., Farràs, O., A.Manjón, J.: Contributory broadcast encryption with efficient encryption and short ciphertexts. IEEE Transactions on Computers 65(2) (2016) 466-479
42. Zhang, L., Wu, Q., Domingo-Ferrer, J., Qin, B., Zeng, P.: Signatures in hierarchical certificateless cryptography: Efficient constructions and provable security. Information Sciences 272 (2014) 223-237
43. Zhang, M., Yang, B., Takagi, T.: Anonymous spatial encryption under affine space delegation functionality with full security. Information Sciences 277 (2014) 715-730
