Direct Chosen-Ciphertext Secure Hierarchical ID-Based Encryption Schemes*

Jong Hwan Park and Dong Hoon Lee

Center for Information Security Technologies(CIST),
Korea University, Seoul, Korea
{decartian,donghlee}@korea.ac.kr

Abstract. We describe two Hierarchical Identity Based Encryption (HIBE) schemes which are selective-ID chosen ciphertext secure. Our constructions are based on the Boneh-Boyen and the Boneh-Boyen-Goh HIBE schemes respectively. We apply the signature-based method to their HIBE schemes. The proposed l-level HIBE schemes are directly derived from l-level HIBE schemes secure against chosen plaintext attacks without padding on identities with one-bit. This is more compact than the known generic transformation suggested by Canetti et al..

Keywords: Hierarchical Identity Based Encryption, Chosen Ciphertext Security.

1 Introduction

Hierarchical Identity Based Encryption (HIBE) [17,16,4,5] is a generalization of Identity Based Encryption (IBE) [18,7,19,15] which allows a sender to encrypt a message for a receiver using the receiver's identity as a public key. In an l-level HIBE scheme, an identity is represented as ID-vectors of length at most l, and a private key for identity at depth $k(< l)$ can be used to derive private keys of its descendant identities. HIBE schemes could be applied to design forward-secure encryption schemes [12,20], and to convert a broadcast encryption scheme in the symmetric key setting into a public key broadcast encryption scheme [14]. Recently, Boyen et al. [11] suggested an anonymous HIBE scheme which mainly gives several application in the public key encryption with keyword search [1].

To prove the security for HIBE schemes without random oracles, Canetti et al. [12] defined a weaker security model called selective-ID security model, and proposed a HIBE scheme. Their scheme is selective-ID secure without random oracles, but that is not efficient. Later, Boneh and Boyen [4] provided an efficient HIBE (denoted by BB_1) scheme, and thereafter Boneh, Boyen, and Goh [5] presented an improved HIBE (denoted by BBG) scheme where the number of

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)(IITA-2006-(C1090-0603-0025)).

J. Lopez, P. Samarati, and J.L. Ferrer (Eds.): EuroPKI 2007, LNCS 4582, pp. 94–109, 2007.

ciphertext elements and pairing operations are independent of the hierarchy depth. These two HIBE schemes suggested by Boneh et al. were provably secure in the selective-ID model without random oracles. More recently, the techniques of constructing the BB_1 and BBG schemes were combined with a public key broadcast encryption scheme [8] in order to achieve the forward security [2].

Chosen ciphertext security of the BB_1 and BBG schemes are obtained from the generic transformation, proposed by Canetti, Halevi, and Katz [13]. The CHK transformation enables construction of an l-level HIBE scheme selective-ID secure against chosen ciphertext attacks based on any $(l+1)$-level HIBE scheme selective-ID secure against chosen plaintext attacks. The CHK transformation, improved upon by [9,10], is generic and extended to the case of adaptive-ID security model (i.e., the full security model) [6].

The CHK transformation requires one-time signature scheme to check the consistency of ciphertext. The important point is that a verification key associated with the one-time signature needs to be embedded into ciphertext in encryption procedure. For this, the authors [13] add one level to an identity hierarchy and set the verification key as an identity. Thus, the CHK transformation considered an $(l+1)$-level HIBE scheme as a subroutine in constructing an l-level HIBE scheme secure against chosen ciphertext attacks. We notice that the CHK transformation needs extra one-bit padding on identities, due to their security proof.

In this paper we construct two HIBE schemes which are provably secure against chosen ciphertext attacks in the selective-ID model. Two schemes are based on the the BB_1 and BBG schemes respectively. We apply the idea of the CHK transformation to their schemes, using one-time signature. At first sight, our constructions appear to apply the CHK transformation to the BB_1 and BBG schemes, but we obtain chosen ciphertext security of l-level HIBE schemes from l-level HIBE schemes secure against chosen plaintext attacks *directly*, without padding on identities with one-bit. Though our approach is not generic, that could be also applied to the concrete schemes [2] with structures of the BB_1 and BBG schemes.

The important algebraic property for security proofs is the one introduced by Boneh et al. [4]. Briefly speaking, for random elements g_1 and g_2 in $\mathbb{G}$ (where $\mathbb{G}$ is generated by a generator g), and random elements r_1, r_2, and r_3 in $\mathbb{Z}_p$ (where r_1 must be non-zero), we have that

$$g_2^{-r_2/r_1}(g_1^{r_1}g^{r_2})^{r_3} = g_2^u(g_1^{r_1}g^{r_2})^{r_3-v/r_1}$$

where $u = \log_g g_1$ and $v = \log_g g_2$. For example, if we let $g_1 = g^a$ and $g_2 = g^b$, the value g_2^u becomes g^{ab}, and if we let $g_1 = g^\alpha$ and $g_2 = g^{\alpha^l}$, the value g_2^u becomes $g^{\alpha^{l+1}}$. The former plays a central role of proving the security of our first construction based on the BB_1 scheme, and the latter does in proving the security of our second construction based on the BBG scheme.

2 Preliminaries

We briefly review the definition of security for HIBE. We also summarize the bilinear maps and the related security assumptions.

2.1 Selective-ID Security Model for HIBE

In a Hierarchical Identity Based Encryption (HIBE) scheme [16,4,5], identities are considered as vectors. That is, an identity of depth l is a tuple $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_l)$. A HIBE scheme consists of the four algorithms [4,5]: *Setup*, *KeyGen*, *Encrypt*, *Decrypt*. The *Setup* algorithm generates system parameters *params* and a master key *master-key*. The *KeyGen* algorithm takes as input an identity $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_l)$ at depth l and the private key $d_{\mathrm{ID}|l-1}$ of the parent identity $\mathrm{ID}_{|l-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{l-1})$ at depth $l-1$. It outputs the private key d_{ID} for identity ID. To encrypt messages, the *Encrypt* algorithm requires a receiver's identity (as a public key) and the system parameters. The *Decrypt* algorithm decrypts ciphertexts with a private key associated with the receiver's identity.

To prove the chosen ciphertext security for HIBE schemes without random oracles, we are interested in the selective-ID security model suggested by Canetti et al. [12,13]. This model is weaker than the full security model (for HIBE schemes, see [5]) in that, in the selective-ID model the adversary commits ahead of time to the identity that it wishes to be challenged on. Since Canetti et al. first proposed the selective-ID model, many cryptographic protocols [4,5,8,11] were proved secure in this weaker security model without random oracles. Selective-ID security model for HIBE schemes is defined via the following game between an adversary $\mathcal{A}$ and a challenger:

Init: $\mathcal{A}$ outputs an identity ID^* where it wishes to be challenged.

Setup: The challenger runs *Setup* algorithm. It gives $\mathcal{A}$ the resulting system parameters *params*. It keeps the *master-key* to itself.

Phase 1: $\mathcal{A}$ issues queries $q_1, ..., q_m$ adaptively where query q_i is one of:
- Private key query on ID_i where $\mathrm{ID}_i \neq \mathrm{ID}^*$ and ID_i is not a prefix of ID^*. The challenger responds by running *KeyGen* algorithm to generate the private key d_i corresponding to the public key ID_i. It sends d_i to $\mathcal{A}$.
- Decryption query CT_i on ID^* or any prefix of ID^*. The challenger responds by running *KeyGen* algorithm to generate the private key d corresponding to ID^*. It then runs *Decrypt* algorithm to decrypt the ciphertext CT_i using the private key d and sends the resulting plaintext to $\mathcal{A}$.

Challenge: Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ on which it wishes to be challenged. The challenger picks a random bit $b \in \{0, 1\}$ and computes $\mathsf{CT} = \mathit{Encrypt}(M_b, \mathit{params}, \mathrm{ID}^*)$ as the challenge ciphertext. It sends CT as the challenge to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues more queries $q_{m+1}, ..., q_n$ adaptively where q_i is one of:
- Private key query on ID_i where $\mathrm{ID}_i \neq \mathrm{ID}^*$ and ID_i is not a prefix of ID^*. The challenger responds as in Phase 1.

- Decryption query $\mathsf{CT}_i \neq \mathsf{CT}$ on ID* or any prefix of ID*. The challenger responds as in Phase 1.

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

We refer to such an adversary $\mathcal{A}$ as an IND-sID-CCA adversary. The advantage of $\mathcal{A}$ in breaking the HIBE scheme $\mathcal{E}$ is defined as

$$\mathrm{Adv}_{\mathcal{E},\mathcal{A}} = \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

Definition 1. *We say that a HIBE scheme $\mathcal{E}$ is $(t, q_{ID}, q_C, \epsilon)$-selective-ID, adaptive chosen ciphertext secure if for any t-time IND-sID-CCA adversary $\mathcal{A}$ that makes at most q_{ID} chosen private key queries, at most q_C chosen decryption queries we have that $Adv_{\mathcal{E},\mathcal{A}} < \epsilon$.*

2.2 Complexity Assumptions

We briefly summarize the bilinear maps, and review the Bilinear Diffie-Hellman (BDH) and the Bilinear Diffie-Hellman Exponent (BDHE) assumptions.

Bilinear Groups: We follow the notation in [7,4].

1. $\mathbb{G}$ and $\mathbb{G}_1$ are two (multiplicative) cyclic groups of prime order p.
2. g be a generator of $\mathbb{G}$.
3. e is a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$.

A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ has the following properties:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ as above. Note that $e(,)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

Bilinear Diffie-Hellman Assumption: The BDH problem in $\mathbb{G}$ is defined as follows: given a tuple $(g, g^a, g^b, g^c) \in \mathbb{G}^4$ as input, compute $e(g, g)^{abc} \in \mathbb{G}_1$. An algorithm $\mathcal{A}$ has advantage ϵ in solving BDH in $\mathbb{G}$ if

$$\Pr\left[\mathcal{A}(g, g^a, g^b, g^c) = e(g, g)^{abc}\right] \geq \epsilon$$

where the probability is over the random choice of a,b,c in $\mathbb{Z}_p$ and the random bits of $\mathcal{A}$. We can also say that an algorithm $\mathcal{B}$ that outputs $b \in \{0, 1\}$ has advantage ϵ in solving the *decision* BDH problem in $\mathbb{G}$ if

$$\left|\Pr\left[\mathcal{B}(g, g^a, g^b, g^c,\ e(g, g)^{abc}) = 0\right] - \Pr\left[\mathcal{B}(g, g^a, g^b, g^c,\ T) = 0\right]\right| \geq \epsilon$$

where the probability is over the random choice of a,b,c in $\mathbb{Z}_p$, the random choice of $T \in \mathbb{G}_1$, and the random bits of $\mathcal{B}$.

Definition 2. *We say that the (decision) (t, ϵ)-BDH assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the (decision) BDH problem in $\mathbb{G}$.*

Bilinear Diffie-Hellman Exponent Assumption: The l-BDHE problem in $\mathbb{G}$ is defined as follows: given a $(2l+1)$-tuple $(g, h, g^x, \ldots, g^{x^l}, g^{x^{l+2}}, \ldots, g^{x^{2l}}) \in \mathbb{G}^{2l+1}$ as input, compute $e(g,h)^{x^{l+1}} \in \mathbb{G}_1$. An algorithm $\mathcal{A}$ has advantage ϵ in solving q-BDHE in $\mathbb{G}$ if

$$\Pr\Big[\mathcal{A}(g, h, g^x, \ldots, g^{x^l}, g^{x^{l+2}}, \ldots, g^{x^{2l}}) = e(g,h)^{x^{l+1}}\Big] \geq \epsilon$$

where the probability is over the random choice of x in $\mathbb{Z}_p$, the random choice of $h \in \mathbb{G}$, and the random bits of $\mathcal{A}$. Let $\overrightarrow{g}_{x,l} = (g^x, \ldots, g^{x^l}, g^{x^{l+2}}, \ldots, g^{x^{2l}})$. Similarly, we say that an algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving the *decision* q-BDHE problem in $\mathbb{G}$ if

$$\Big|\Pr\Big[\mathcal{B}(g, h, \overrightarrow{g}_{x,l},\ e(g,h)^{x^{l+1}}) = 0\Big] - \Pr\Big[\mathcal{B}(g, h, \overrightarrow{g}_{x,l},\ T) = 0\Big]\Big| \geq \epsilon$$

where the probability is over the random choice of x in $\mathbb{Z}_p$, the random choice of $h \in \mathbb{G}$, the random choice of $T \in \mathbb{G}_1$, and the random bits of $\mathcal{B}$.

Definition 3. *We say that the (decision) (t, l, ϵ)-BDHE assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the (decision) l-BDHE problem in $\mathbb{G}$.*

3 Chosen Ciphertext Secure HIBE from the BB_1 Scheme

In this section we present an l-level HIBE scheme that is derived from the l-level BB_1 scheme, using the idea of the CHK transformation. The constructed l-level HIBE scheme is secure against chosen ciphertext attacks in the selective-ID model without random oracles. For the CHK transformation, we need a one-time signature scheme $Sig = (SigKeyGen, Sign, Verify)$ which is strongly existentially unforgeable (see the details in [3]). We also need a collision resistant hash function that maps verification keys to $\mathbb{Z}_p$. For simplicity, we assume that the verification keys are elements of $\mathbb{Z}_p$.

3.1 Construction

Setup(k): To generate HIBE system parameters for maximum depth of l, select random $\alpha \in \mathbb{Z}_p^*$ and set $g_1 = g^\alpha$. Next, pick random elements $h, h_1, \ldots, h_l \in \mathbb{G}$ and a generator $g_2 \in \mathbb{G}$. The public parameters *params* (with the description of $(\mathbb{G}, \mathbb{G}_1, p)$) and the secret *master-key* are given by

$$params = (g, g_1, g_2, h, h_1, \ldots, h_l), \qquad master\text{-}key = g_2^\alpha.$$

For $j = 1, \ldots, l$, define $F_j : \mathbb{Z}_p \to \mathbb{G}$ to be the function: $F_j(x) = g_1^x h_j$.

KeyGen($d_{\mathrm{ID}|j-1}$, ID): To create a private key d_{ID} for a user $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in \mathbb{Z}_p^j$ of depth $j \leq l$, pick random $r_1, \ldots, r_j \in \mathbb{Z}_p$ and output

$$d_{\mathrm{ID}} = \Big(g_2^{\alpha} \prod_{k=1}^{j} F_k(\mathrm{I}_k)^{r_k},\ g^{r_1}, \ldots, g^{r_j} \Big).$$

The private key for ID can be also generated from a private key for $d_{\mathrm{ID}|j-1}$. Let $d_{\mathrm{ID}|j-1} = (d_0, \ldots, d_{j-1})$ be the private key for $\mathrm{ID}_{j-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{j-1})$. After selecting random $r_1, \ldots, r_j \in \mathbb{Z}_p$, output d_{ID} as

$$\Big(d_0 \cdot \prod_{k=1}^{j} F_k(\mathrm{I}_k)^{r_k},\ d_1 \cdot g^{r_1}, \ldots,\ d_{j-1} \cdot g^{r_{j-1}},\ g^{r_j} \Big).$$

Encrypt(M, *params*, ID): To encrypt a message $M \in \mathbb{G}_1$ under a public key $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in \mathbb{Z}_p^j$,

1. Run the *SigKeyGen* to obtain a signing key SigK and a verification key VerK.
2. Pick a random $s \in \mathbb{Z}_p^*$ and compute
$$C = \Big(g^s,\ e(g_1, g_2)^s \cdot M,\ F_1(I_1)^s, \ldots,\ F_j(I_j)^s,\ (g_1^{\mathsf{VerK}} h)^s \Big).$$
3. Output the ciphertext $\mathsf{CT} = (C, Sign_{\mathsf{SigK}}(C), \mathsf{VerK})$.

Decrypt(CT, *params*, d_{ID}): To decrypt a ciphertext $\mathsf{CT} = (C, \sigma, \mathsf{VerK})$ using the private key $d_{\mathrm{ID}} = (d_0, \ldots, d_j)$,

1. Verify that the signature σ on C is valid under the verification key VerK. If invalid, output $\perp$.
2. Otherwise, let $C = (A, B, C_1, \ldots, C_{j+1})$. Pick a random $r_{j+1} \in \mathbb{Z}_p^*$ and output
$$\frac{\prod_{k=1}^{j} e(C_k,\ d_k) \cdot e(C_{j+1}, g^{r_{j+1}})}{e(A,\ d_0 \cdot (g_1^{\mathsf{VerK}} h)^{r_{j+1}})} \cdot B.$$

The correctness of decryption algorithm is checked as below:

$$\frac{\prod_{k=1}^{j} e(C_k,\ d_k) \cdot e(C_{j+1}, g^{r_{j+1}})}{e(A,\ d_0 \cdot (g_1^{\mathsf{VerK}} h)^{r_{j+1}})} = \frac{\prod_{k=1}^{j} e(F_k(I_k)^s,\ g^{r_k}) \cdot e((g_1^{\mathsf{VerK}} h)^s,\ g^{r_{j+1}})}{e(g^s,\ g_2^{\alpha} \prod_{k=1}^{j} F_k(I_k)^{r_k} \cdot (g_1^{\mathsf{VerK}} h)^{r_{j+1}})}$$
$$= \frac{\prod_{k=1}^{j} e(F_k(I_k)^{r_k},\ g^s) \cdot e((g_1^{\mathsf{VerK}} h)^{r_{j+1}},\ g^s)}{e(g^s,\ g_2^{\alpha}) \cdot e(g^s, \prod_{k=1}^{j} (F_k(I_k))^{r_k} \cdot (g_1^{\mathsf{VerK}} h)^{r_{j+1}})} = \frac{1}{e(g_1, g_2)^s}.$$

At a first glance, the above scheme has a similar structure as the $(l+1)$-level HIBE scheme in that the additional element $h \in \mathbb{G}$ adds to the public parameters and the size of ciphertext increases by one more element. However, the private key for ID is still generated at level $(l-1)$ and is the same as that of chosen plaintext secure l-level HIBE scheme. We note that unlike the BB_1 scheme [4], randomization in the *KeyGen* (in deriving the private keys from its parent identity) and the *Decrypt* algorithms is necessary for the proof of security.

3.2 Security

Theorem 1. *Suppose that the decision (t, ϵ_1)-BDH assumption holds in $\mathbb{G}$ and the signature scheme is $(t, 1, \epsilon_2)$-strongly existentially unforgeable. Then the previous l-HIBE scheme is $(t', q_{\mathrm{ID}}, q_C, \epsilon)$-selective-ID, adaptive chosen ciphertext secure for arbitrary q_{ID}, q_C, and $t' < t - o(t)$, where $\epsilon_1 + q_{\mathrm{ID}}/p + \epsilon_2 \geq \epsilon$.*

Proof. Suppose there exists an adversary $\mathcal{A}$ which has advantage ϵ in attacking the l-level HIBE scheme. We want to build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the decision BDH problem in $\mathbb{G}$. On input (g, g^a, g^b, g^c, T) for some unknown $a, b, c \in \mathbb{Z}_p^*$, $\mathcal{B}$ outputs 1 if $T = e(g, g)^{abc}$ and 0 otherwise. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective-ID game as follows:

Init: $\mathcal{A}$ outputs an identity $\mathrm{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*) \in \mathbb{Z}_p^k$ of depth $k \leq l$ that it intends to attack.

Setup: Let $g_1 = g^a$, $g_2 = g^b$, and $g_3 = g^c$. If the length of ID^* is less than l, $\mathcal{B}$ selects random elements $(\mathrm{I}_{k+1}^*, \ldots, \mathrm{I}_l^*)$ in $\mathbb{Z}_p$. To generate the system parameters, $\mathcal{B}$ first selects random $\alpha_1, \ldots, \alpha_l \in \mathbb{Z}_p$ and defines $h_j = g_1^{-\mathrm{I}_j^*} g^{\alpha_j}$ for $j = 1, \ldots, l$. Next, $\mathcal{B}$ runs *SigKeyGen* algorithm to gain a signing key SigK^* and a verification key VerK^*, and $\mathcal{B}$ also selects a random $\beta \in \mathbb{Z}_p$ and computes $h = g_1^{-\mathsf{VerK}^*} g^{\beta}$. $\mathcal{B}$ gives $\mathcal{A}$ the system parameters $params = (g, g_1, g_2, h, h_1, \ldots, h_l)$. The master key corresponding to these $params$ is $g_2^a = g^{ab}$, which is unknown to $\mathcal{B}$. For $j = 1, \ldots, l$, the function $F_j : \mathbb{Z}_p \to \mathbb{G}$ is defined as

$$F_j(x) = g_1^x h_j = g_1^{x - \mathrm{I}_j^*} g^{\alpha_j}.$$

Phase 1: $\mathcal{A}$ issues up to q_{ID} private key queries and q_C decryption queries. Consider a query for the private key corresponding to $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_u) \in \mathbb{Z}_p^u$ where $u \leq l$. We further distinguish two cases according to whether ID^* is not a prefix of ID or not.

First, consider the case ID^* is not a prefix of ID. Then there exists at least one $j \in \{1, \ldots, u\}$ such that $\mathrm{I}_j \neq \mathrm{I}_j^*$. To respond to the query, $\mathcal{B}$ responds to the query by first computing a private key for the identity $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$ from which it derives a private key for the requested identity $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j, \ldots, \mathrm{I}_u)$. $\mathcal{B}$ picks random elements $r_1, \ldots, r_j \in \mathbb{Z}_p$ and computes

$$d_0 = g_2^{\frac{-\alpha_j}{\mathrm{I}_j - \mathrm{I}_j^*}} \prod_{v=1}^{j} F_v(\mathrm{I}_v)^{r_v}, \quad d_1 = g^{r_1}, \ldots, \; d_{j-1} = g^{r_{j-1}}, \; d_j = g_2^{\frac{-1}{\mathrm{I}_j - \mathrm{I}_j^*}} g^{r_j}.$$

By the same argument as in [4], we see that $(d_0, d_1, \ldots, d_j)$ is a valid private key for $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$. For the unknown $\widetilde{r}_j = r_j - b/(\mathrm{I}_j - \mathrm{I}_j^*)$, $\mathcal{B}$ has

$$g_2^{\frac{-\alpha_j}{\mathrm{I}_j - \mathrm{I}_j^*}} F_j(\mathrm{I}_j)^{r_j} = g_2^{\frac{-\alpha_j}{\mathrm{I}_j - \mathrm{I}_j^*}} (g_1^{\mathrm{I}_j - \mathrm{I}_j^*} g^{\alpha_j})^{r_j} = g_2^a F_j(\mathrm{I}_j)^{\widetilde{r}_j}, \qquad d_j = g^{\widetilde{r}_j}.$$

Then, $\mathcal{B}$ can construct a private key for the requested ID from the above private key $(d_0, d_1, \ldots, d_j)$ and gives $\mathcal{A}$ the obtained private key d_{ID}.

Second, consider the case ID^* is a prefix of ID. Then it satisfies that $k+1 \leq u$. Let $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_u)$. If $\mathrm{I}_j = \mathrm{I}_j^*$ for $j = k+1, \ldots, u$, then $\mathcal{B}$ outputs a random bit $b \in \{0,1\}$ and aborts the simulation. Otherwise, there exists at least one $j \in \{k+1, \ldots, u\}$ such that $\mathrm{I}_j \neq \mathrm{I}_j^*$. $\mathcal{B}$ responds to the query by first computing a private key for $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_j)$ from which it constructs a private key for the requested $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_j, \ldots, \mathrm{I}_u)$. $\mathcal{B}$ picks random elements $r_1, \ldots, r_j \in \mathbb{Z}_p$. Let $\widetilde{r}_j = r_j - b/(\mathrm{I}_j - \mathrm{I}_j^*)$. Then $\mathcal{B}$ generates the private key for $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_j)$ as

$$d_0 = g_2^{\frac{-\alpha_j}{\mathrm{I}_j - \mathrm{I}_j^*}} \prod_{v=1}^{k} F_v(\mathrm{I}_v)^{r_v}, \quad d_1 = g^{r_1}, \ \ldots, \ d_k = g^{r_k},$$

$$d_{k+1} = g^{r_{k+1}}, \ \ldots, \ d_j = g_2^{\frac{-1}{\mathrm{I}_j - \mathrm{I}_j^*}} g^{r_j}.$$

By the similar argument above, this private key has a proper distribution and is computable.

Next, $\mathcal{B}$ responds to decryption queries for $\mathrm{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$ or any prefix of ID^*. Let $\mathrm{ID}' = (\mathrm{I}_1^*, \ldots, \mathrm{I}_j^*)$ where $j \leq k$ and let $(C, \sigma, \mathsf{VerK})$ be a decryption query for ID' where $C = (A, B, C_1, \ldots, C_{j+1})$. $\mathcal{B}$ does as follows:

1. Run $Verify$ to check the validity of the signature σ on C, using the verification key VerK. If the signature is invalid, $\mathcal{B}$ responds with $\perp$.
2. If $\mathsf{VerK} = \mathsf{VerK}^*$, $\mathcal{B}$ outputs a random bit $b \in \{0,1\}$ and aborts the simulation.
3. Otherwise, $\mathcal{B}$ selects random $\{r_i\}$ for $i = 1, \ldots, j+1$, and computes

$$\widetilde{d}_0 = g_2^{\frac{-\beta}{\mathsf{VerK} - \mathsf{VerK}^*}} (g_1^{\mathsf{VerK} - \mathsf{VerK}^*} g^{\beta})^{r_{j+1}} \cdot \prod_{v=1}^{j} F_v(\mathrm{I}_v^*)^{r_v},$$

$$\widetilde{d}_1 = g^{r_1}, \ \ldots, \ \widetilde{d}_j = g^{r_j}, \ \widetilde{d}_{j+1} = g_2^{\frac{-1}{\mathsf{VerK} - \mathsf{VerK}^*}} g^{r_{j+1}}.$$

   As the above, for some (unknown) $\widetilde{r}_{j+1} = r_{j+1} - b/(\mathsf{VerK} - \mathsf{VerK}^*)$, we see that

$$g_2^{\frac{-\beta}{\mathsf{VerK} - \mathsf{VerK}^*}} (g_1^{\mathsf{VerK} - \mathsf{VerK}^*} g^{\beta})^{r_{j+1}} = g_2^{a} (g_1^{\mathsf{VerK} - \mathsf{VerK}^*} g^{\beta})^{\widetilde{r}_{j+1}} = g_2^{a} (g_1^{\mathsf{VerK}} h)^{\widetilde{r}_{j+1}},$$

   and $\widetilde{d}_{j+1} = g^{\widetilde{r}_{j+1}}$. Then, $\mathcal{B}$ computes the plaintext as

$$\frac{\prod_{v=1}^{j} e(C_v, \ \widetilde{d}_v) \cdot e(C_{j+1}, \widetilde{d}_{j+1})}{e(A, \ \widetilde{d}_0)} \cdot B.$$

This computation is identical to the *Decrypt* algorithm in a real attack, since $\{r_i\}$ for $i = 1, \ldots, j+1$ are uniform in $\mathbb{Z}_p$ and $\widetilde{d}_0 = g_2^{a} \cdot \prod_{v=1}^{j} F_v(\mathrm{I}_v^*)^{r_v} \cdot (g_1^{\mathsf{VerK}} h)^{\widetilde{r}_{j+1}}$.

Challenge: $\mathcal{A}$ outputs two messages $M_0, M_1 \in \mathbb{G}_1$. To encrypt one of the two messages under the public key ID^*, $\mathcal{B}$ selects a random bit $b \in \{0,1\}$ and computes $C = (g_3,\ M_b \cdot T,\ g_3^{\alpha_1}, \ldots,\ g_3^{\alpha_j},\ g_3^{\beta})$. Next, $\mathcal{B}$ gives the challenge ciphertext $\mathsf{CT} = (C, Sign_{\mathsf{SigK}^*}(C), \mathsf{VerK}^*)$ to $\mathcal{A}$. Since $F_i(\mathrm{I}_i^*) = g^{\alpha_i}$ for $i = 1, \ldots, j$ and $g_1^{\mathsf{VerK}^*} h = g^{\beta}$, we have that

$$C = (g^c,\ M_b \cdot T,\ F_1(\mathrm{I}_1^*)^c, \ldots,\ F_l(\mathrm{I}_l^*)^c,\ (g_1^{\mathsf{VerK}^*} h)^c).$$

If $T = e(g,g)^{abc} = e(g_1, g_2)^c$, then C is a valid encryption of M_b under the public key ID^*. Otherwise, $M_b \cdot T$ is just a random element of $\mathbb{G}_1$ and independent of the bit b in the adversary's view.

Phase 2: $\mathcal{A}$ issues more private key and decryption queries. $\mathcal{B}$ responds as in Phase 1.

Guess : $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b = b'$ then $\mathcal{B}$ outputs 1, indicating $T = e(g,g)^{abc}$. Otherwise, it outputs 0, indicating $T \neq e(g,g)^{abc}$.

We consider two cases. When T is random in $\mathbb{G}_1$ then $\Pr[\mathcal{B}(g, g^a, g^b, g^c, T) = 0] = 1/2$. Let Iden denote the event that $\mathcal{A}$ issues a private key query for $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_u)$ such that $\mathrm{I}_j = \mathrm{I}_j^*$ for $i = k+1, \ldots, u$. Also, let Forge denote the event that $\mathcal{A}$ submits a valid ciphertext $\mathsf{CT} = (C, \sigma, \mathsf{VerK}^*)$ as a decryption query. In the cases of Iden and Forge, $\mathcal{B}$ cannot reply to the private key and decryption queries, and aborts the simulation. When $T = e(g,g)^{abc}$, $\mathcal{B}$ replied with valid private key and plaintext unless events Iden and Forge occur. Then, $\mathcal{B}$ has

$$\left|\Pr[\mathcal{B}(g, g^a, g^b, g^c, T) = 0] - \frac{1}{2}\right| \geq \left|\Pr[b = b' \wedge \overline{\mathsf{Iden}} \wedge \overline{\mathsf{Forge}}] - \frac{1}{2}\right| - \Pr[\mathsf{Iden}] - \Pr[\mathsf{Forge}].$$

Since $\mathcal{B}$ provided $\mathcal{A}$ with perfect simulation when events Iden and Forge did not occur, $|\Pr[b = b' \wedge \overline{\mathsf{Iden}} \wedge \overline{\mathsf{Forge}}] - 1/2| \geq \epsilon$. From the simple calculation, we know that $\Pr[\mathsf{Iden}]$ is at most q_{ID}/p. Also, note that $\Pr[\mathsf{Forge}]$ is negligible. This means that $\Pr[\mathsf{Forge}] < \epsilon_2$ since otherwise, $\mathcal{B}$ can construct a forger, which is contradiction to the one-time signature. Therefore,

$$\left|\Pr\Big[\mathcal{B}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0\Big] - \Pr\Big[\mathcal{B}(g, g^a, g^b, g^c, T) = 0\Big]\right| \geq \epsilon - \frac{q_{\mathrm{ID}}}{p} - \epsilon_2$$

This completes the proof of Theorem 1. □

4 Chosen Ciphertext Secure HIBE from the BBG Scheme

We present an l-level HIBE scheme secure against chosen ciphertext attacks based on the l-level BBG scheme secure against chosen plaintext attacks. As in the previous section, we need a one-time signature scheme $Sig = (SigKeyGen, Sign, Verify)$, and we assume that verifications keys are elements of $\mathbb{Z}_p$.

4.1 Construction

Setup(k): To generate public parameters for maximum depth of l, select random $\alpha \in \mathbb{Z}_p^*$ and set $g_1 = g^\alpha$. Next, pick random elements $g_2, g_3, v, h_1, \ldots, h_l \in \mathbb{G}$. The public parameters *params* (with the description of $(\mathbb{G}, \mathbb{G}_1, p)$) and the secret *master-key* are given by

$$params = (g, g_1, g_2, g_3, v, h_1, \ldots, h_l), \qquad master\text{-}key = g_4 = g_2^\alpha$$

KeyGen($d_{\mathrm{ID}|j-1}$, ID): To create a private key d_{ID} for a user $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in \mathbb{Z}_p^j$ of depth $j \le l$, pick random $r \in \mathbb{Z}_p$ and output

$$d_{\mathrm{ID}} = \Big(g_2^\alpha \cdot (h_1^{\mathrm{I}_1} \cdots h_j^{\mathrm{I}_j} \cdot g_3)^r,\ g^r,\ v^r,\ h_{j+1}^r, \ldots,\ h_l^r\Big).$$

The private key for ID can be also generated from a private key for $d_{\mathrm{ID}|j-1}$. Let

$$\begin{aligned} d_{\mathrm{ID}|j-1} &= \Big(g_2^\alpha \cdot (h_1^{\mathrm{I}_1} \cdots h_{j-1}^{\mathrm{I}_{j-1}} \cdot g_3)^{r'},\ g^{r'},\ v^{r'},\ h_j^{r'}, \ldots,\ h_l^{r'}\Big) \\ &= (a_0, a_1, a_2, b_j, \ldots, b_l\ ) \end{aligned}$$

be the private key for $\mathrm{ID}_{|j-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{j-1}) \in \mathbb{Z}_p^{j-1}$. To generate d_{ID}, pick a random $r^* \in \mathbb{Z}_p$ and output

$$d_{\mathrm{ID}} = \Big(a_0 \cdot b_j^{\mathrm{I}_j} \cdot (h_1^{\mathrm{I}_1} \cdots h_j^{\mathrm{I}_j} \cdot g_3)^{r^*},\ a_1 \cdot g^{r^*},\ a_2 \cdot v^{r^*},\ b_{j+1} \cdot h_{j+1}^{r^*}, \ldots,\ b_l \cdot h_l^{r^*}\Big).$$

Since $r = r' + r^*$, we see that this private key is a properly distributed private key for $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$.

Encrypt(M, *params*, ID): To encrypt a message $M \in \mathbb{G}_1$ under a public key $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j) \in \mathbb{Z}_p^j$,

1. Run the *SigKeyGen* to obtain a signing key SigK and a verification key VerK.
2. Pick a random $s \in \mathbb{Z}_p^*$ and compute

$$C = \Big(g^s,\ e(g_1, g_2)^s \cdot M,\ (h_1^{\mathrm{I}_1} \cdots h_j^{\mathrm{I}_j} \cdot v^{\mathsf{VerK}} \cdot g_3)^s\Big).$$

3. Output the ciphertext $\mathsf{CT} = (C, Sign_{\mathsf{SigK}}(C), \mathsf{VerK})$.

Decrypt(CT, *params*, d_{ID}): Consider an identity $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$. To decrypt a ciphertext $\mathsf{CT} = (C, \sigma, \mathsf{VerK})$ using the private key $d_{\mathrm{ID}} = (a_0, a_1, a_2, b_{j+1}, \ldots, b_l)$,

1. Check that the signature σ on C is valid under the key VerK. If invalid, output $\perp$.
2. Otherwise, let $C = (C_1, C_2, C_3)$. Select a random $w \in \mathbb{Z}_p$ and compute

$$\widetilde{a}_0 = a_0 \cdot a_2^{\mathsf{VerK}} \cdot (h_1^{\mathrm{I}_1} \cdots h_j^{\mathrm{I}_j} \cdot v^{\mathsf{VerK}} \cdot g_3)^w, \qquad \widetilde{a}_1 = a_1 \cdot g^w.$$

3. Output $\big(e(C_1,\ \widetilde{a}_1)/e(C_3,\ \widetilde{a}_0)\big) \cdot C_2$.

Note that the pair $(\widetilde{a}_0,\ \widetilde{a}_1)$ is chosen from the following distribution

$$\left(\ g_2^{\alpha}\cdot(h_1^{\mathrm{I}_1}\cdots h_j^{\mathrm{I}_j}\cdot v^{\mathsf{VerK}}\cdot g_3)^{\widetilde{r}},\quad g^{\widetilde{r}}\ \right)$$

where $\widetilde{r}$ is uniform in $\mathbb{Z}_p$. This distribution is independent of $\mathrm{ID}=(\mathrm{I}_1,\ldots,\mathrm{I}_j)$. Next, the correctness of decryption algorithm is checked as below:

$$\frac{e(C_1,\ \widetilde{a}_1)}{e(C_3,\ \widetilde{a}_0)}=\frac{e((h_1^{\mathrm{I}_1}\cdots h_j^{\mathrm{I}_j}\cdot v^{\mathsf{VerK}}\cdot g_3)^s,\ g^{\widetilde{r}})}{e(g^s,\ g_2^{\alpha}\cdot(h_1^{\mathrm{I}_1}\cdots h_j^{\mathrm{I}_j}\cdot v^{\mathsf{VerK}}\cdot g_3)^{\widetilde{r}})}=\frac{1}{e(g^s,g_2^{\alpha})}=\frac{1}{e(g_1,g_2)^s}.$$

4.2 Security

As opposed to the l-BDHE assumption for the $\mathsf{IND\text{-}sID\text{-}CPA}$ secure BBG scheme in [5], security of the $\mathsf{IND\text{-}sID\text{-}CCA}$ secure HIBE scheme above is based on the $(l+1)$-BDHE assumption.

Theorem 2. *Suppose that the decision $(t, l{+}1, \epsilon_1)$-BDHE assumption holds in $\mathbb{G}$ and the signature scheme is $(t, 1, \epsilon_2)$-strongly existentially unforgeable. Then the previous l-HIBE scheme is $(t', q_{\mathrm{ID}}, q_C, \epsilon)$-selective-ID, adaptive chosen ciphertext secure for arbitrary q_{ID}, q_C, and $t' < t - \Theta(\tau l q_{\mathrm{ID}})$, where $\epsilon_1+\epsilon_2 \geq \epsilon$ and τ is the maximum time for an exponentiation in $\mathbb{G}$.*

Proof. Suppose there exists an adversary $\mathcal{A}$ which has advantage ϵ in attacking the l-level HIBE scheme. We want to build an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the decision $(l+1)$-BDHE problem in $\mathbb{G}$. For a generator $g\in\mathbb{G}$ and $\alpha\in\mathbb{Z}_p$, let $y_i = g^{\alpha^i}\in\mathbb{G}$. On input $(g, h, y_1, \ldots, y_{l+1}, y_{l+3}, \ldots, y_{2l+2}, T)$, $\mathcal{B}$ outputs 1 if $T = e(g,h)^{\alpha^{l+2}}$ and 0 otherwise. $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective-ID game as follows:

Init: $\mathcal{A}$ outputs an identity $\mathrm{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)\in\mathbb{Z}_p^k$ of depth $k\leq l$ that it intends to attack.

Setup: To generate the system parameters, $\mathcal{B}$ first selects random $\rho,\eta\in\mathbb{Z}_p$ and sets $g_1 = y_1 = g^{\alpha}$, $g_2 = y_{l+1}\cdot g^{\rho}$, and $v = y_{l+1}^{\eta}$. Next, $\mathcal{B}$ runs *SigKeyGen* algorithm to gain a signing key SigK^* and a verification key VerK^*. Next, $\mathcal{B}$ picks random $\gamma,\gamma_1,\ldots,\gamma_l$ in $\mathbb{Z}_p$, and sets $h_i = g^{\gamma_i}y_i$ for $i=1,\ldots,l$ and $g_3 = g^{\gamma}\cdot v^{-\mathsf{VerK}^*}\cdot(h_1^{\mathrm{I}_1^*}\ldots, h_k^{\mathrm{I}_k^*})^{-1}$.
Then, it gives $\mathcal{A}$ the system parameters $params = (g, g_1, g_2, g_3, v, h_1, \ldots, h_l)$. The master key corresponding to these $params$ is $g_2^{\alpha} = y_{l+2}\cdot y_1^{\rho}$, which is unknown to $\mathcal{B}$.

Phase 1: $\mathcal{A}$ issues up to q_{ID} private key queries and q_C decryption queries. First, consider a query for the private key corresponding to $\mathrm{ID} = (\mathrm{I}_1,\ldots,\mathrm{I}_u)\in\mathbb{Z}_p^u$ where $u\leq l$. The only restriction is that ID is not a prefix of ID^*. We further distinguish two cases according to whether ID^* is a prefix of ID or not. First, consider the case ID^* is not a prefix of ID. Then there exists $j\in\{1,\ldots,k\}$ such that $\mathrm{I}_j\neq\mathrm{I}_j^*$. To respond to the query, $\mathcal{B}$ first derives a private key for the identity $(\mathrm{I}_1, \ldots, \mathrm{I}_j)$ from which it constructs a private key for the requested identity $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j, \ldots, \mathrm{I}_u)$.

$\mathcal{B}$ picks a random $s \in \mathbb{Z}_p$. Let $\widetilde{s} = s + \alpha^{(l+2-j)}/(\mathrm{I}_j^* - \mathrm{I}_j)$. Next, $\mathcal{B}$ generates the private key for $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_u)$ as

$$\left(g_2^{\alpha} \cdot (h_1^{\mathrm{I}_1} \cdots h_j^{\mathrm{I}_j} \cdot g_3)^{\widetilde{s}},\ g^{\widetilde{s}},\ v^{\widetilde{s}},\ h_{j+1}^{\widetilde{s}}, \ldots,\ h_l^{\widetilde{s}}\right)$$

which is a properly distributed private key for the identity $\mathrm{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$. We show that $\mathcal{B}$ can compute all elements of this private key given the values that it knows. To generate the first component of the private key, observe that

$$\begin{aligned}
(h_1^{\mathrm{I}_1} \cdots h_j^{\mathrm{I}_j} \cdot g_3)^{\widetilde{s}} &= (h_1^{\mathrm{I}_1} \cdots h_j^{\mathrm{I}_j} \cdot g^{\gamma} \cdot v^{-\mathsf{VerK}^*} \cdot h_1^{-\mathrm{I}_1^*} \cdots h_j^{-\mathrm{I}_j^*} \cdots h_k^{-\mathrm{I}_k^*})^{\widetilde{s}} \\
&= (g^{\gamma} \cdot v^{-\mathsf{VerK}^*} \cdot h_j^{\mathrm{I}_j - \mathrm{I}_j^*} \cdot h_{j+1}^{-\mathrm{I}_{j+1}^*} \cdots h_k^{-\mathrm{I}_k^*})^{\widetilde{s}} \\
&= h_j^{\widetilde{s} \cdot (\mathrm{I}_j - \mathrm{I}_j^*)} \cdot (g^{\gamma} \cdot v^{-\mathsf{VerK}^*} \cdot h_{j+1}^{-\mathrm{I}_{j+1}^*} \cdots h_k^{-\mathrm{I}_k^*})^{\widetilde{s}}.
\end{aligned}$$

Note that the value $h_j^{\widetilde{s} \cdot (\mathrm{I}_j - \mathrm{I}_j^*)}$ in the above becomes $y_{l+2}^{-1} \cdot y_j^{s(\mathrm{I}_j - \mathrm{I}_j^*)} \cdot g^{\widetilde{s} \cdot \gamma_j \cdot (\mathrm{I}_j - \mathrm{I}_j^*)}$. Since $g_2^{\alpha} = y_{l+2} \cdot y_1^{\rho}$, the first component can be computed as

$$y_1^{\rho} \cdot y_j^{s(\mathrm{I}_j - \mathrm{I}_j^*)} \cdot g^{\widetilde{s} \cdot \gamma_j \cdot (\mathrm{I}_j - \mathrm{I}_j^*)} \cdot (g^{\gamma} \cdot v^{-\mathsf{VerK}^*} \cdot h_{j+1}^{-\mathrm{I}_{j+1}^*} \cdots h_k^{-\mathrm{I}_k^*})^{\widetilde{s}}$$

where the unknown term y_{l+2} is canceled out. The other terms $g^{\widetilde{s}}$, $v^{\widetilde{s}}$, and $h_i^{\widetilde{s}}$ for $i = j+1, \ldots, k$ are computable since $g^{\widetilde{s}} = g^s \cdot y_{l+2-j}^{1/(\mathrm{I}_j - \mathrm{I}_j^*)}$, $v^{\widetilde{s}} = v^s \cdot y_{2l+3-j}^{\eta/(\mathrm{I}_j - \mathrm{I}_j^*)}$, and $h_i^{\widetilde{s}} = g^{\gamma_i \cdot s} \cdot y_{l+2-j}^{\gamma_i/(\mathrm{I}_j^* - \mathrm{I}_j)} \cdot y_i^s \cdot y_{l+2-j+i}^{1/(\mathrm{I}_j^* - \mathrm{I}_j)}$ for $i = j+1, \ldots, k$. These values do not require knowledge of y_{l+2}. Similarly, the remaining elements $g^{\widetilde{s}}$, $h_{j+1}^{\widetilde{s}}, \ldots, h_l^{\widetilde{s}}$ can be computed since they do not involve the y_{l+2} term.

Second, consider the case when ID^* is a prefix of ID. Then it holds that $k+1 \le u$. Let $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_u)$. In this step, we can assume that there exists at least one $j \in \{k+1, \ldots, u\}$ such that $\mathrm{I}_j \neq 0$ in $\mathbb{Z}_p$. Otherwise, for all $j \in \{k+1, \ldots, u\}$, $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, 0, \ldots, 0)$. Then this private key for ID can be easily used to decrypt the challenge ciphertext. Let j be the smallest index such that $\mathrm{I}_j \neq 0$. $\mathcal{B}$ responds to the query by first computing a private key for $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_j)$ from which it constructs a private key for the requested $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \mathrm{I}_{k+1}, \ldots, \mathrm{I}_j, \ldots, \mathrm{I}_u)$. $\mathcal{B}$ selects a random $s \in \mathbb{Z}_p$. Let $\widetilde{s} = s - \alpha^{(l+2-j)}/\mathrm{I}_j$. Then $\mathcal{B}$ generates the private key for $\mathrm{ID} = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*, \ldots, \mathrm{I}_j)$ as

$$\left(g_2^{\alpha} \cdot (h_1^{\mathrm{I}_1^*} \cdots h_k^{\mathrm{I}_k^*} \cdots h_j^{\mathrm{I}_j} \cdot g_3)^{\widetilde{s}},\ g^{\widetilde{s}},\ v^{\widetilde{s}},\ h_{j+1}^{\widetilde{s}}, \ldots,\ h_l^{\widetilde{s}}\right).$$

By the similar argument above, this private key has a proper distribution and is computable.

Next, $\mathcal{B}$ responds to decryption queries for $\mathrm{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$ or any prefix of ID^*. Let $\mathrm{ID}' = (\mathrm{I}_1^*, \ldots, \mathrm{I}_j^*)$ where $j \leq k$ and let $(C, \sigma, \mathsf{VerK})$ be a decryption query for ID' where $C = (C_1, C_2, C_3)$. $\mathcal{B}$ does as follows:

1. Run $Verify$ to check the validity of the signature σ on C, using the verification key VerK. If the signature is invalid, $\mathcal{B}$ responds with $\perp$.
2. If $\mathsf{VerK} = \mathsf{VerK}^*$, $\mathcal{B}$ outputs a random bit $b \in \{0,1\}$ and aborts the simulation.
3. Otherwise, $\mathcal{B}$ checks that the equality $e(h_1^{\mathrm{I}_1^*} \ldots h_j^{\mathrm{I}_j^*} \cdot v^{\mathsf{VerK}} \cdot g_3, C_1) \stackrel{?}{=} e(C_2, g)$. If it does not hold, $\mathcal{B}$ knows that (C_1, C_2) is not of the right form. Then, $\mathcal{B}$ outputs a random message $M \in \mathbb{G}_1$. Otherwise, for some (unknown) $s \in \mathbb{Z}_p$ such that $C_1 = g^s$, $\mathcal{B}$ has that $C_2 = (h_1^{\mathrm{I}_1^*} \ldots h_j^{\mathrm{I}_j^*} \cdot v^{\mathsf{VerK}} \cdot g_3)^s$. Plugging in the value of g_3, C_2 becomes

$$\begin{aligned} C_2 &= \left(h_1^{\mathrm{I}_1^*} \ldots h_j^{\mathrm{I}_j^*} \cdot v^{\mathsf{VerK}} \cdot g^\gamma \cdot v^{-\mathsf{VerK}^*} \cdot h_1^{-\mathrm{I}_1^*} \ldots h_k^{-\mathrm{I}_k^*}\right)^s \\ &= \left(v^{\mathsf{VerK}-\mathsf{VerK}^*} \cdot g^\gamma \cdot h_{j+1}^{-\mathrm{I}_{j+1}^*} \ldots h_k^{-\mathrm{I}_k^*}\right)^s \\ &= \left(y_{l+1}^{\eta(\mathsf{VerK}-\mathsf{VerK}^*)} \cdot g^\gamma\right)^s \cdot \left(h_{j+1}^{-\mathrm{I}_{j+1}^*} \ldots h_k^{-\mathrm{I}_k^*}\right)^s. \end{aligned}$$

$\mathcal{B}$ computes $\widetilde{a}_0 = y_1^{-\gamma/\eta(\mathsf{VerK}-\mathsf{VerK}^*)} \cdot C_2 \cdot (h_{j+2}^{-\mathrm{I}_{j+1}^*} \ldots h_{k+1}^{-\mathrm{I}_k^*})^{-1/\eta(\mathsf{VerK}-\mathsf{VerK}^*)}$ and $\widetilde{a}_1 = C_1 \cdot y_1^{-1/\eta(\mathsf{VerK}-\mathsf{VerK}^*)}$. Let $\widetilde{r} = s - \alpha/\eta(\mathsf{VerK} - \mathsf{VerK}^*)$. Then,

$$\begin{aligned} \widetilde{a}_0 &= y_1^{-\gamma/\eta(\mathsf{VerK}-\mathsf{VerK}^*)} \cdot \left(y_{l+1}^{\eta(\mathsf{VerK}-\mathsf{VerK}^*)} \cdot g^\gamma\right)^s \cdot \left(h_{j+1}^{-\mathrm{I}_{j+1}^*} \ldots h_k^{-\mathrm{I}_k^*}\right)^{\widetilde{r}} \\ &= y_{l+2} \cdot \left(y_{l+1}^{\eta(\mathsf{VerK}-\mathsf{VerK}^*)} \cdot g^\gamma\right)^{\widetilde{r}} \cdot \left(h_{j+1}^{-\mathrm{I}_{j+1}^*} \ldots h_k^{-\mathrm{I}_k^*}\right)^{\widetilde{r}} \\ &= y_{l+2} \cdot \left(v^{\mathsf{VerK}} \cdot g^\gamma \cdot v^{-\mathsf{VerK}^*} \cdot h_{j+1}^{-\mathrm{I}_{j+1}^*} \ldots h_k^{-\mathrm{I}_k^*}\right)^{\widetilde{r}}, \\ \widetilde{a}_1 &= g^s \cdot y_1^{-1/\eta(\mathsf{VerK}-\mathsf{VerK}^*)} = g^{\widetilde{r}}. \end{aligned}$$

Recall that the master-key is $y_{l+2} \cdot y_1^\rho$. For the re-randomization, $\mathcal{B}$ selects a random $r' \in \mathbb{Z}_p$ and computes $\widetilde{a}_0' = \widetilde{a}_0 \cdot y_1^\rho \cdot (v^{\mathsf{VerK}} \cdot g^\gamma \cdot v^{-\mathsf{VerK}^*} \cdot h_{j+1}^{-\mathrm{I}_{j+1}^*} \ldots h_k^{-\mathrm{I}_k^*})^{r'}$ and $\widetilde{a}_1' = \widetilde{a}_1 \cdot g^{r'}$. For some (unknown) $\widetilde{r}' = \widetilde{r} + r'$,

$$\begin{aligned} \widetilde{a}_0' &= y_{l+2} \cdot y_1^\rho \cdot \left(v^{\mathsf{VerK}} \cdot g^\gamma \cdot v^{-\mathsf{VerK}^*} \cdot h_{j+1}^{-\mathrm{I}_{j+1}^*} \ldots h_k^{-\mathrm{I}_k^*}\right)^{\widetilde{r}} \\ &= g_2^\alpha \cdot \left(h_1^{\mathrm{I}_1^*} \cdots h_j^{\mathrm{I}_j^*} \cdot v^{\mathsf{VerK}} \cdot g_3\right)^{\widetilde{r}'}, \\ \widetilde{a}_1' &= g^{\widetilde{r}} \cdot g^{r'} = g^{\widetilde{r}'}. \end{aligned}$$

$\mathcal{B}$ responds with $\left(e(C_1, \widetilde{a}_1')/e(C_3, \widetilde{a}_0')\right) \cdot C_2$. This response is identical to *Decrypt* algorithm in a real attack, because r' (and $\widetilde{r}'$) is uniform in $\mathbb{Z}_p$.

Challenge: $\mathcal{A}$ outputs two messages $M_0, M_1 \in \mathbb{G}_1$. To encrypt one of the two messages under the public key ID*, $\mathcal{B}$ selects a random bit $b \in \{0,1\}$ and a random $t \in \mathbb{Z}_p$. $\mathcal{B}$ computes $C = (h^t,\ T^t \cdot e(y_1, h)^{t\cdot\rho} \cdot M_b,\ h^{t\cdot\gamma})$, where T and h are from the input tuple given to $\mathcal{B}$. Next, $\mathcal{B}$ gives the challenge ciphertext $\mathsf{CT} = (C, Sign_{\mathsf{SigK}^*}(C), \mathsf{VerK}^*)$ to $\mathcal{A}$. If $h = g^c$ for some (unknown) $c \in \mathbb{Z}_p$, $h^{t\cdot\gamma} = (h_1^{\mathrm{I}_1^*} \cdots h_k^{\mathrm{I}_k^*} \cdot v^{\mathsf{VerK}} \cdot g_3)^{t\cdot c}$. Define $\mu = t \cdot c \in \mathbb{Z}_p$. On the one hand, if $T = e(g,h)^{\alpha^{l+2}}$, we have that

$$C = \left(g^{\mu},\ e(g_1, g_2)^{\mu} \cdot M_b,\ (h_1^{\mathrm{I}_1^*} \cdots h_k^{\mathrm{I}_k^*} \cdot v^{\mathsf{VerK}^*} \cdot g_3)^{\mu}\right)$$

which is a valid encryption of M_b under the public key ID$^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_k^*)$. On the other hand, when T is uniform and independent in $\mathbb{G}_1$, then C (and CT) is independent of b in the adversary's view.

Phase 2: $\mathcal{A}$ issues more private key and decryption queries. $\mathcal{B}$ responds as in Phase 1.

Guess : $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b = b'$ then $\mathcal{B}$ outputs 1, indicating $T = e(g,h)^{\alpha^{l+2}}$. Otherwise, it outputs 0, indicating $T \neq e(g,h)^{\alpha^{l+2}}$.

When T is random in $\mathbb{G}_1$ then $\Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,l+1}, T) = 0] = 1/2$. Let Forge denote the event that $\mathcal{A}$ submits a valid ciphertext $\mathsf{CT} = (C, \sigma, \mathsf{VerK}^*)$ as a decryption query. In the case of Forge, $\mathcal{B}$ cannot reply to the decryption query and aborts the simulation. When $T = e(g,h)^{\alpha^{l+2}}$, $\mathcal{B}$ replied with a valid plaintext unless event Forge occurs. Then, $\mathcal{B}$ has

$$\left|\Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,l+1}, T) = 0] - \frac{1}{2}\right| \geq \left|\Pr[b = b' \wedge \overline{\mathsf{Forge}}] - \frac{1}{2}\right| - \Pr[\mathsf{Forge}].$$

Since $\mathcal{B}$ provided $\mathcal{A}$ with perfect simulation when event Forge did not occur, $|\Pr[b = b' \wedge \overline{\mathsf{Forge}}] - 1/2| \geq \epsilon$. Also, note that $\Pr[\mathsf{Forge}]$ is negligible. This means that $\Pr[\mathsf{Forge}] < \epsilon_2$ since otherwise, $\mathcal{B}$ can construct a forger, which is contradiction to the one-time signature. Therefore,

$$\left|\Pr\Big[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,l+1}, e(g,g)^{abc}) = 0\Big] - \Pr\Big[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,l+1}, T) = 0\Big]\right| \geq \epsilon - \epsilon_2$$

This completes the proof of Theorem 1. □

5 Conclusion

We presented two HIBE schemes that are secure against chosen ciphertext attacks in the selective-ID model, based on the BB_1 and BBG schemes. We obtain chosen ciphertext security of the l-level HIBE schemes by directly applying the idea of the CHK transformation to the l-level BB_1 and BBG schemes. The resulting schemes are more compact than the ones derived from the known generic transformation for chosen ciphertext secure l-level HIBE scheme.

Moreover, our constructions imply that the CHK transformation could be applied to obtain chosen ciphertext security of concrete schemes with the BB_1 and BBG-like structures.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extentions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)
2. Attrapadung, N., Furukawa, J., Imai, H.: Forward-secure and searchable broadcast encryption with short ciphertexts and private keys. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 161–177. Springer, Heidelberg (2006)
3. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
4. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
5. Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
6. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Journal (submission) Available from `http://crypto.stanford.edu/~dabo/pubs.html`
7. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
8. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
9. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A.J. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
10. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM Conference on Computer and Communications Security - CCS'05, pp. 320–329. ACM Press, New York (2005)
11. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
12. Canetti, C., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRPYT 2003. LNCS, vol. 2656, Springer, Heidelberg (2003)
13. Canetti, C., Halevi, S., Katz, J.: Chosen ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
14. Dodis, Y., Fazio, N.: Public key broadcast encryption for stateless receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)

15. Gentry, C.: Practical Identity-Based Encryption Without Random Oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
16. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
17. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
18. Shamir, A.: Identity-based cryptosystems and signaure shcemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
19. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
20. Yao, D., Fazio, N., Dodis, Y., Lysyanskaya, A.: ID-based encryption for complex hierarchies with applications to forward security and broadcast encrypption. In: ACM Conference on Computer and Communications Security - CCS'04, pp. 354–363 (2004)