Anonymous Hierarchical Identity-Based Encryption with Constant Size Ciphertexts

Jae Hong Seo[1,⋆], Tetsutaro Kobayashi[2], Miyako Ohkubo[2], and Koutarou Suzuki[2]

[1] Department of Mathematical Sciences and ISaC-RIM, Seoul National University, Seoul, Korea
`jhsbhs@gmail.com`
[2] NTT Information Sharing Platform Labs, Tokyo, Japan
{`kobayashi.tetsutaro,ookubo.miyako,suzuki.koutarou`}`@lab.ntt.co.jp`

Abstract. We propose an anonymous Hierarchical Identity-Based Encryption (anonymous HIBE) scheme that has constant size ciphertexts. This means the size of the ciphertext does not depend on the depth of the hierarchy. Moreover, our scheme achieves the lowest computational cost because during the decryption phase the computational cost of decryption is constant. The security can be proven under reasonable assumptions without using random oracles because it is based on the composite order bilinear group. Our scheme achieves selective-ID security notion.

1 Introduction

Identity-Based Encryption (IBE) is a topic of focus as a useful technique. Studies are proceeding in various directions, and numerous applications using IBE have been presented. A searchable encryption scheme has been discussed. At first, schemes allowed keywords that were not encrypted. However, in such schemes, an anonymous request for a keyword cannot be satisfied by using simple IBE schemes. To provide this function, anonymous IBE was proposed. The anonymous IBE scheme provides a very useful function, i.e., anonymity of ID. An anonymous IBE ciphertext does not leak any information about the receiver's identity. Such a useful property can be applied to keyword searchable encryption while maintaining anonymity of the keyword [1,5,11].

Anonymous Hierarchical Identity-Based Encryption (anonymous HIBE), which handles IDs hierarchically maintaining the anonymity of an ID and keys, can be delegated even if a blinding ID is used. Anonymous HIBE allows some protocols using anonymous HIBE to be extended; for example, by applying keyword-searchable encryption, keywords can be treated hierarchically while maintaining anonymous keyword information.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-642-00468-1_29

⋆ This work was done while the first author was visiting NTT Information Sharing Platform Labs. He was supported by the Science Research Center(SRC) program of the Korea Science and Engineering Foundation (KOSEF) with a grant funded by the Korean govement (Ministry of Science and Technology (MOST), Grant R11-2007-035-01002-0).

1.1 Related Works: ID-Based Encryption Algorithms

After Horwitz and Lynn defined the notion of Hierarchical ID-Based Encryption (HIBE)[18], many efficient and provably secure HIBE schemes were proposed. Gentry and Silverberg proposed an efficient and secure HIBE scheme, that achieves full-ID CPA (chosen plaintext attack) security; however, it was proven with a random oracle (GS-HIBE)[17]. Canetti, Halevi and Katz[13] suggested a weaker security notion, called selective-ID, and they also proposed a selective-ID secure HIBE without using random oracles; however, their scheme is an inefficient one. An efficient and selective-ID secure HIBE scheme in the standard model was proposed by Boneh and Boyen (BB-HIBE) [2]. However the ciphertext of the BB-HIBE scheme is depends on the depth of the hierarchy. To improve the efficiency, HIBE with constant size ciphertexts was presented by Boneh, Boyen and Goh (BBG-HIBE)[3]. In their scheme, a private key can be delegated while maintaining a constant ciphertext size. BBG-HIBE was proven without using random oracles, and it achieves selective-ID security. Full-ID secure schemes that do not use random oracles were presented by Waters[23], Chatterjee and Sarkar [15]. All of the above mentioned HIBE schemes were proposed assuming that IDs are known to everyone, so they cannot provide anonymity of ID. We call such HIBE schemes non-anonymous HIBE schemes.

On the other hand, the concepts of anonymous IBE and anonymous HIBE were shown by Abdalla et al. [1], and formal definitions of them were also given in that paper. However, a concrete construction of anonymous HIBE was not proposed. A concrete constructions of anonymous IBE in the standard model was proposed by Gentry[16] and a concrete construction of anonymous HIBE was proposed by Boyen and Waters (BW-HIBE)[12]. Both of these schemes were proposed in the standard model and are selective-ID CPA secure, which can be proven without using random oracles. Shi and Waters proposed a delegatable hidden-vecor encryption (dHVE) whose definition is a generalization of anonymous HIBE, i.e., anonymous HIBE is a special case of the dHVE scheme (SW-dHVE) [22]. The scheme takes a composit order bilinear group, which was introduced by Boneh, Goh, and Nissim [7], to obtain property of anonoymity. However, in the BW-HIBE and SW-dHVE schemes, the ciphertext size depends on the hierarchy depth. The ciphertext size has a great impact on practicality, so while their results are very interesting, efficiency remains an open problem.

1.2 Our Results

Motivation: The size of ciphertext affects the efficiency and feasibility of various applications using HIBE schemes, and if the ciphertext size depends on the depth the hierarchy depth, the efficiency and feasibility of applications also depend on the hierarchy depth. The HIBE scheme with constant size ciphertexts can extend the feasibility and convenience of applications. Additionally, some applications need anonymity of ID. For example, Keyword-search encryption is one of such applications. In keyword-search encryption scheme, each keyword needs a ciphertext, therefore totally the size of ciphertexts for many keywords is significant impact on the efficiency of keyword search. In such a case the size of ciphertext is serious problem. However, none of the previous results for anonymous HIBE could provide constant size ciphertexts.

Contribution: We present an anonymous HIBE scheme with constant size ciphertexts. Our scheme achieves selective-ID CPA security without using random oracles, and is based on a new assumption, the ℓ-composite Diffie-Hellman assumption The details are shown in section 2.3. The technical highlight of our paper is the technique for achieving constant size ciphertexts even though the number of layers is increased in HIBE, keeping anonymity of IDs, and security proofs using game-based proof techniques. The difficulties in devising an efficient anonymous HIBE scheme are constructing a length of ciphertext that is independent of hierarchy depth and maintaining the anonymity of key delegation hierarchical ID. Our idea is effected by BBG-HIBE [3], which provides constant size ciphertexts. However, [3] does not satisfy the requirement of an anonymous ID. To attain ID anonymity, we need a randomizing method, keeping the property of key deligation. Therefore the proposed scheme takes a composite order bilinear group, and is using a technique improved upon that of in [11]. The HVE scheme in [11] is not anonymous HIBE; it is a scheme for keyword searchable encryption. However, the technique for providing keyword anonymity offers us a key idea for solving key delegation for anonymous HIBE.

There were two anonymous HIBE schemes, BW-HIBE and SW-dHVE[1] before our HIBE scheme. Comparing our construction with these previous two anonymous HIBE schemes, we see that the ciphertexts in both those schemes are $O(L)$ group elements, and private keys are $O(L^2)$ group elements, where L is the maximum hierarchy depth. In constract, our scheme uses only four group elements for a ciphertext, and the private key uses $O(L)$ group elements.

2 Background

2.1 Security Models

We briefly explain the informal security notions of anonymous HIBE. The formal security definitions may be found in the literature [6,1]. We use a weaker notion of security introduced by [13,14] in which the adversary commits ahead of time to the public *params* that it will attack; i.e., we use the selective security notion.

Semantic security(IND-sID(indistinguishability against selective identity)): The adversary outputs target identity ID* before public parameters are generated. It can make a private key derivation query for ID such that the ID is not a prefix of or equal to target identity ID*. It publishes target message Msg*. No poly-time adversary can distinguish between a ciphertext of target message Msg* with target identity ID* and a ciphertext of random message with target identity ID*.

[1] The dHVE and anonymous HIBE schemes were proposed as anonymous HIBE schemes [22]. Anonymous HIBE has a flaw, but since we can consider dHVE as an anonymous HIBE scheme, we compare dHVE with our scheme.

Anonymity(ANON-sID(anonymity against selective identity)): The adversary outputs target identity ID* before public parameters are generated. It can make private key derivation query for ID such that the ID is not a prefix of or equal to target identity ID*. It publishes target message Msg*. No poly-time adversary can distinguish between a ciphertext of target message Msg* with target identity ID* and a ciphertext of target message Msg* with random identity.

2.2 Bilinear Groups of Composite Order

We will use a bilinear group of composite order pq. Bilinear groups of composite order were introduced by Boneh, Goh, Nissim [7].

Let $\mathcal{G}$ be a group generation algorithm that takes security parameter 1^λ as input and outputs tuple $(p, q, \mathbb{G}, \mathbb{G}_T, e)$ where p and q are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $n = pq$, and e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a non-degenerate bilinear map; i.e., e satisfies the following properties:

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

We assume that group multiplication in $\mathbb{G}$, $\mathbb{G}_T$ and bilinear map e are all polynomial time computable in λ. Furthermore, we assume that descriptions of $\mathbb{G}$ and $\mathbb{G}_T$ contain generators as well as identity elements $1_{\mathbb{G}}$, $1_{\mathbb{G}_T}$ of $\mathbb{G}$ and $\mathbb{G}_T$, respectively. If there is no confusion, we use 1 for identity irrespective of the group.

We will use the notation $\mathbb{G}_p$ and $\mathbb{G}_q$ to denote the subgroups of $\mathbb{G}$ of order p and q, respectively, and we will use the notation $\mathbb{G}_{T,p}$ and $\mathbb{G}_{T,q}$ to denote subgroups of $\mathbb{G}_T$ of order p and q, respectively. Then $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q$ and $\mathbb{G}_T = \mathbb{G}_{T,p} \times \mathbb{G}_{T,q}$. If g_1 is a generator of $\mathbb{G}$, then g_1^q and g_1^p are generators of $\mathbb{G}_p$ and $\mathbb{G}_q$, respectively. We use the notation g_p and g_q to denote generators of $\mathbb{G}_p$ and $\mathbb{G}_q$, respectively.

Note that $e(h_p, h_q) = 1$ for all random elements $h_p \in \mathbb{G}_p$ and $h_q \in \mathbb{G}_q$ because $e(h_p, h_q) = e(g_p^a, g_q^b)$ for some integers a, b, and $e(g_p^a, g_q^b) = e(g_1^{qa}, g_1^{pb}) = e(g_1, g_1)^{pqab} = 1$ for some generator g_1 in $\mathbb{G}$.

2.3 Complexity Assumptions

ℓ-weak Bilinear Diffie-Hellman Inversion* assumption. The ℓ-Bilinear Diffie-Hellman Inversion (ℓ-BDHI) assumption has been used for constructing cryptographic schemes [21,2,3,4]. Boneh, Boyen and Goh introduced a slightly weaker assumption, ℓ-weak BDHI*, denoted by ℓ-wBDHI* to design HIBE with constant size ciphertexts in [4]. Our scheme use Decision ℓ-wBDHI* in bilinear groups of composite order to prove semantic security. We say that group generator $\mathcal{G}$ satisfies the (ϵ, t)-Decision ℓ-wBDHI* assumption if no t-time algorithm has advantage at least ϵ in solving the Decision ℓ-wBDHI* problem in groups generated by $\mathcal{G}$.

$$(prime : \{p, q\}, group : \{\mathbb{G}, \mathbb{G}_T\}, bilinearmap : \{e\}) \overset{R}{\leftarrow} \mathcal{G}(\lambda),$$

$$n \leftarrow pq, \quad g_p, h \xleftarrow{R} \mathbb{G}_p, \quad g_q \xleftarrow{R} \mathbb{G}_q, \quad a \xleftarrow{R} \mathbb{Z}_n$$

$$\mathbf{Z} \leftarrow ((n, \mathbb{G}, \mathbb{G}_T, e), g_q, g_p, h, g_p^{a}, g_p^{a^2}, \cdots, g_p^{a^\ell}),$$

$$T \leftarrow e(g_p, h)^{a^{\ell+1}}, \quad d \xleftarrow{R} \{0, 1\}.$$

Let $T' = T$ if d is 1; otherwise set T' to be a uniformly and independently chosen element from $\mathbb{G}_{T,p}$. We call $(\mathbf{Z}, T')$ the challenge pair of the Decision ℓ-wBDHI*. Give the challenge pair to adversary $\mathcal{A}$. Then $\mathcal{A}$ outputs d', and succeeds if $d = d'$. The advantage of $\mathcal{A}$ in solving Devision ℓ-wBDHI* problem in groups generated by $\mathcal{G}$ is $|Pr[\mathcal{A}(\mathbf{Z}, T) = 1] - Pr[\mathcal{A}(\mathbf{Z}, R) = 1]|$, where the probability is over random coins in $\mathcal{G}$, a random choice of $R \in \mathbb{G}_{T,p}$ and the random coins of $\mathcal{A}$.

ℓ-composite Diffie-Hellman assumption. The anonymity of our construction is based on a new complexity assumption that we call the ℓ-composite Diffie-Hellman assumption (ℓ-cDH) in bilinear groups with composite order $n = pq$. We say that group generator $\mathcal{G}$ satisfies the (ϵ, t)-ℓ-cDH assumption if no t-time algorithm has advantage at least ϵ in solving the ℓ-cDH problem in groups generated by $\mathcal{G}$.

$$(prime : \{p, q\}, group : \{\mathbb{G}, \mathbb{G}_T\}, bilinearmap : \{e\}) \xleftarrow{R} \mathcal{G}(\lambda),$$

$$n \leftarrow pq, \quad g_p \xleftarrow{R} \mathbb{G}_p, \quad g_q, R_1, R_2, R_3 \xleftarrow{R} \mathbb{G}_q \quad a, b \xleftarrow{R} \mathbb{Z}_n$$

$$\mathbf{Z} \leftarrow ((n, \mathbb{G}, \mathbb{G}_T, e), g_q, g_p, g_p^{a}, g_p^{a^2}, \cdots, g_p^{a^\ell}, g_p^{a^{\ell+1}} \cdot R_1, g_p^{a^{\ell+1}b} \cdot R_2),$$

$$T \leftarrow g_p^{b} \cdot R_3, \quad d \xleftarrow{R} \{0, 1\}.$$

Let $T' = T$ if d is 1; otherwise set T' to be a uniformly and independently chosen element from $\mathbb{G}$. We call $(\mathbf{Z}, T')$ the challenge pair of the ℓ-cDH. Give the challenge pair to adversary $\mathcal{A}$. Then $\mathcal{A}$ outputs d', and succeeds if $d = d'$. The advantage of $\mathcal{A}$ in solving the ℓ-cDH problem in groups generated by $\mathcal{G}$ is $|Pr[\mathcal{A}(\mathbf{Z}, T) = 1] - Pr[\mathcal{A}(\mathbf{Z}, R) = 1]|$, where the probability is over random coins in $\mathcal{G}$, a random choice of $R \in \mathbb{G}$ and the random coins of $\mathcal{A}$.

Our assumption holds in a generic model if the factorization of n is hard. According to the Master Theorem [20], in a generic model, if there is an algorithm $\mathcal{A}$ issuing at most q_s instructions and having advantage Adv in the above experiment, then $\mathcal{A}$ can be used to find a non-trivial factor of n with probability at least $Adv - O(q_s^2(\ell + 2)/(n^{1/2}))$. Therefore, if the factorization of n is hard, any polynomial time algorithm $\mathcal{A}$ has a negligible advantage in n.

Bilinear Subset Decision assumption. This assumption is implied by the ℓ-cDH assumption and was introduced by Boneh, Sahai, and Waters [10]. We say that group generator $\mathcal{G}$ satisfies the (ϵ, t)-Bilinear Subset Decision (BSD) assumption if no t-time algorithm has advantage at least ϵ in solving the BSD problem in groups generated by $\mathcal{G}$.

$$(prime : \{p, q\}, group : \{\mathbb{G}, \mathbb{G}_T\}, bilinearmap : \{e\}) \xleftarrow{R} \mathcal{G}(\lambda),$$

$$n \leftarrow pq, \quad g_p \overset{R}{\leftarrow} \mathbb{G}_p, \quad g_q \overset{R}{\leftarrow} \mathbb{G}_q,$$
$$\mathbf{Z} \leftarrow ((n, \mathbb{G}, \mathbb{G}_T, e), g_q, g_p),$$
$$T \overset{R}{\leftarrow} \mathbb{G}_{T,p}, \quad d \overset{R}{\leftarrow} \{0, 1\}.$$

Let $T' = T$ if d is 1; otherwise set T' to be a uniformly and independently chosen element from $\mathbb{G}_T$. We call $(\mathbf{Z}, T')$ the challenge pair of the BSD problem. Give the challenge pair to adversary $\mathcal{A}$. Then $\mathcal{A}$ outputs d', and succeeds if $d = d'$. The advantage of $\mathcal{A}$ in solving BSD problem in groups generated by $\mathcal{G}$ is $|Pr[\mathcal{A}(\mathbf{Z}, T) = 1] - Pr[\mathcal{A}(\mathbf{Z}, R) = 1]|$, where the probability is over random coins in $\mathcal{G}$, a random choice of $R \in \mathbb{G}_T$ and the random coins of $\mathcal{A}$.

3 Anonymous HIBE with Constant Size Ciphertexts

In this section, we propose an anonymous hierarchical ID-based encryption with constant size ciphertexts secure under the Decision ℓ-wBDH* assumption and ℓ-cDH assumption. Our construction is based on BBG-HIBE. To attain anonymity, we construct HIBE over a bilinear group with composite order as HVE in [11] which can be considered as an anonymous IBE scheme. All non-random elements of our HIBE scheme are embedded in $\mathbb{G}_p$ or $\mathbb{G}_{T,p}$. A private key consists of only elements in $\mathbb{G}_p$. Public parameters and ciphertexts are blinded by random elements in $\mathbb{G}_q$ or $\mathbb{G}_{T,q}$. Since a pairing result between elements in $\mathbb{G}_p$ and elements in $\mathbb{G}_q$ is 1, blinding factors of ciphertexts are removed by calculating a pairing with a private key in the decryption procedure.

In delegation procedure of the BBG-HIBE scheme, the private key is re-randomized by using public parameters. However, we cannot use public parameters for private key re-randomization in our construction since a private key must be composed of only elements in $\mathbb{G}_p$. If we use public parameters which have blinding factors in $\mathbb{G}_q$ to re-randomize the private key, then the resulting private key will also be blinded by elements in $\mathbb{G}_q$. This private key can not decrypt any ciphertext because the blinding factors of ciphertexts cannot be removed by pairing with the private key in the decryption procedure. Therefore we add a re-randomization subkey, which is composed of elements in $\mathbb{G}_p$, to the private key, and the re-randomization procedure uses not only public parameters but also the re-randomization subkey.

3.1 Construction

Here, we present our HIBE construction.

Setup(λ, L)**:** The setup algorithm generates public system parameters, denoted by *params*, and the corresponding master secret key, denoted by *MK* by using a security parameter and the maximum hierarchy depth L. First, the setup algorithm generates $(p, q, \mathbb{G}, \mathbb{G}_T, e)$, as explained in the section 2.2. Next, it selects random elements

$$g, f, v, h_1, \cdots, h_L, w \in \mathbb{G}_p, \qquad R_g, R_f, R_v, R_1, \cdots, R_L, g_q \in \mathbb{G}_q.$$

It then, computes $G = gR_g$, $F = fR_f$, $V = vR_v$, $H_1 = h_1R_1, \cdots, H_L = h_LR_L$ and $E = e(g, w)$, and it publishes the description of a group $\mathbb{G}$ and *params* as:

$$params \leftarrow [g_q, g_p, G, F, V, H_1, \cdots, H_L, E]$$

and retains *MK* as the secret values:

$$MK \leftarrow [p,\ q,\ g,\ f,\ v,\ h_1,\ \cdots,\ h_L,\ w]$$

The group description contains n but not p, q.

KeyGenerate(*MK*, ID)**:** To generate a private key corresponding to $\mathrm{ID} = [\mathrm{I_1}, \mathrm{I_2}, \cdots, \mathrm{I_k}] \in (\mathbb{Z}_n)^k$, the *KeyGenerate* algorithm first takes *MK* and ID as input. Next, it picks random integers $r_1, r_2, s_1, s_2, t_1, t_2 \in \mathbb{Z}_n$ satisfying equations $s_1 \cdot t_2 - s_2 \cdot t_1 \not\equiv 0 \mod p$ and $\not\equiv 0 \mod q$. The algorithm randomly chooses integers and checks whether or not the equation holds. If it does not, the algorithm chooses other random integers and repeats this procedure until the equation does hold. Since the equation holds without probability $\frac{p+q-1}{n}$, this iteration will finish immediately. The private key $\mathrm{Pvk^{ID}}$ consisting of two sub-keys $\mathrm{Pvk_d^{ID}} \in (\mathbb{G}_p)^{L-k+3}$ and $\mathrm{Pvk_r^{ID}} \in (\mathbb{G}_p)^{2(L-k+3)}$ is output. $\mathrm{Pvk_d^{ID}}$ is used for decryption and delegation, and $\mathrm{Pvk_r^{ID}}$ is used for re-randomization.

$$\mathrm{Pvk_d^{ID}} \leftarrow [w(v\prod_{i=1}^{k} h_i^{\mathrm{I_i}})^{r_1} f^{r_2},\ g^{r_1},\ g^{r_2},\ h_{k+1}^{r_1},\ \cdots\ h_L^{r_1}].$$

$$\mathrm{Pvk_r^{ID}} \leftarrow [[(v\prod_{i=1}^{k} h_i^{\mathrm{I_i}})^{s_1} f^{s_2}, g^{s_1}, g^{s_2}, h_{k+1}^{s_1}, \cdots h_L^{s_1}], [(v\prod_{i=1}^{k} h_i^{\mathrm{I_i}})^{t_1} f^{t_2}, g^{t_1}, g^{t_2}, h_{k+1}^{t_1}, \cdots h_L^{t_1}]].$$

Derive($\mathrm{Pvk^{ID_{|k-1}}}, \mathrm{ID_{|k}}$)**:** The private key for $\mathbf{ID_{|k}} \in (\mathbb{Z}_n)^k$, where $2 \le k \le L$, is derived from a given private key for the parent,

$$\mathrm{Pvk^{ID_{|k-1}}} = [\mathrm{Pvk_d^{ID_{|k-1}}}, \mathrm{Pvk_r^{ID_{|k-1}}}]$$

$$= [\ [a_0, a_1, a_2, b_k, \cdots, b_L],\ [\ [\alpha_0, \alpha_1, \alpha_2, \beta_k, \cdots, \beta_L],\ [\alpha'_0, \alpha'_1, \alpha'_2, \beta'_k, \cdots, \beta'_L]\ ]\ ].$$

To generate $\mathrm{Pvk^{ID_{|k}}}$, pick random integers γ_1, γ_2, γ_3, δ_1, δ_2, $\delta_3 \in \mathbf{Z}_n$ satisfying equations $g_p^{\gamma_2 \cdot \delta_3 - \gamma_3 \cdot \delta_2} \not\equiv 1$ and $g_q^{\gamma_2 \cdot \delta_3 - \gamma_3 \cdot \delta_2} \not\equiv 1$ holds. To select four integers satisfying the equations, uniformly and independently choose four integers from $\mathbb{Z}_n$ and check the equation. If the equation does not hold, then choose four other integers and repeat the procedure. Since four randomly chosen integers γ_2, γ_3, δ_2 and δ_3 satisfy the above equations without negligible probability $\frac{p+q-1}{n}$, this iteration will finish immediately. Therefore we consider four randomly chosen integers satisfying above equations as a random element in $GL_2(\mathbb{Z}_n)$. Lastly the *Derive* algorithm outputs $\mathrm{Pvk_d^{ID_{|k}}}$ and $\mathrm{Pvk_r^{ID_{|k}}}$ as follows.

Step 1 (delegation procedure):

$$[\ \zeta_0, \zeta_1, \zeta_2, \eta_{k+1}, \cdots, \eta_L\ ] \leftarrow [\ a_0 \cdot b_k^{\mathrm{I_k}}, a_1, a_2, b_{k+1}, \cdots, b_L\ ]$$

$$[\ \theta_0, \theta_1, \theta_2, \phi_{k+1}, \cdots, \phi_L\ ] \leftarrow [\alpha_0 \cdot \beta_k^{\mathrm{I_k}}, \alpha_1, \alpha_2, \beta_{k+1}, \cdots, \beta_L],$$
$$[\ \theta'_0, \theta'_1, \theta'_2, \phi'_{k+1}, \cdots, \phi'_L\ ] \leftarrow [\alpha'_0 \cdot \beta'^{\mathrm{I_k}}_k, \alpha'_1, \alpha'_2, \beta'_{k+1}, \cdots, \beta'_L].$$

Step 2(re-randomization procedure):

$$\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}|\mathrm{k}} \leftarrow [\ \zeta_0 \theta_0^{\gamma_1} \theta_0'^{\delta_1}, \zeta_1 \theta_1^{\gamma_1} \theta_1'^{\delta_1}, \zeta_2 \theta_2^{\gamma_1} \theta_2'^{\delta_1}, \eta_{k+1} \phi_{k+1}^{\gamma_1} \phi_{k+1}'^{\delta_1}, \cdots, \eta_L \phi_L^{\gamma_1} \phi_L'^{\delta_1}\ ]$$
$$\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}|\mathrm{k}} \leftarrow [\ [\theta_0^{\gamma_2} \theta_0'^{\delta_2}, \theta_1^{\gamma_2} \theta_1'^{\delta_2}, \theta_2^{\gamma_2} \theta_2'^{\delta_2}, \phi_{k+1}^{\gamma_2} \phi_{k+1}'^{\delta_2}, \cdots, \phi_L^{\gamma_2} \phi_L'^{\delta_2}\ ],$$
$$[\theta_0^{\gamma_3} \theta_0'^{\delta_3}, \theta_1^{\gamma_3} \theta_1'^{\delta_3}, \theta_2^{\gamma_3} \theta_2'^{\delta_3}, \phi_{k+1}^{\gamma_3} \phi_{k+1}'^{\delta_3}, \cdots, \phi_L^{\gamma_3} \phi_L'^{\delta_3}]\ ].$$

We note that private keys generated by the *Derive* algorithm have the same structure and distribution as those generated by the *KeyGenerate* algorithm. Two random integers r_1 and r_2 of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}|\mathrm{k}-1}$ are re-randomized as follows:

$$\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} + \begin{pmatrix} s_1 & t_1 \\ s_2 & t_2 \end{pmatrix} \cdot \begin{pmatrix} \gamma_1 \\ \delta_1 \end{pmatrix}$$

Since we choose γ_1 and δ_1 uniformly and independently from $\in \mathbb{Z}_n$, the above value is also distributed uniformly in $(\mathbb{Z}_n)^2$. Therefore, $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}|\mathrm{k}}$ has the same distribution as that of the private key generated by the *KeyGenerate* algorithm.

Random integers of $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}|\mathrm{k}-1}$ are re-randomized as follows:

$$A \cdot B \ \ where \ \ A = \begin{pmatrix} s_1 & t_1 \\ s_2 & t_2 \end{pmatrix}, \ \ B = \begin{pmatrix} \gamma_2 & \gamma_3 \\ \delta_2 & \delta_3 \end{pmatrix}$$

Since $A \in GL_2(\mathbb{Z}_n)$ and B are uniformly chosen from $GL_2(\mathbb{Z}_n)$, $A \cdot B$ is also uniformly distributed in $GL_2(\mathbb{Z}_n)$. Therefore, the private key generated by the *Derive* algorithm has the same distribution as that of the private key generated by the *KeyGenerate* algorithm.

Encrypt$(params, \mathrm{ID}, \mathrm{Msg})$**:** First, pick a random integer $s \in \mathbb{Z}_n$ and random elements $Z_1, Z_2, Z_3 \in \mathbb{G}_q$ to encrypt message $\mathrm{Msg} \in \mathbb{G}_T$ for a given identity $\mathrm{ID} = [\mathrm{I}_1, \cdots, \mathrm{I_k}] \in (\mathbb{Z}_n)^k$. A random element of $\mathbb{G}_q$ can be chosen by raising g_q to random exponents from $\mathbb{Z}_n$. Next, the *Encrypt* algorithm outputs the ciphertext

$$\mathrm{CT} \ \leftarrow \ [\mathrm{Msg} \cdot E^s,\ G^s \cdot Z_1,\ F^s \cdot Z_2,\ (V \prod_{i=1}^{k} H^{\mathrm{I_i}})^s \cdot Z_3] \ \in \mathbb{G}_T \times \mathbb{G}^3.$$

Decrypt$(\mathrm{Pvk}^{\mathrm{ID}}, \mathrm{CT})$**:** Consider $\mathrm{ID} = [\mathrm{I}_1, \cdots, \mathrm{I_k}]$. To decrypt ciphertext $\mathrm{CT} = [C_1, C_2, C_3, C_4]$, using the first three elements of subkey $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}} = [a_0, a_1, a_2, b_{k+1}, \cdots, b_L]$ of the private key $\mathrm{Pvk}^{\mathrm{ID}}$, output

$$\mathrm{Msg} \ \leftarrow \ C_1 \cdot \frac{e(a_1, C_4) \cdot e(a_2, C_3)}{e(a_0, C_2)}.$$

We can easily check the correctness of the *Decrypt* algorithm for valid ciphertext as follows.

$$C_1 \frac{e(a_1, C_4)e(a_2, C_3)}{e(a_0, C_2)} = \mathrm{Msg} \cdot e(g, w)^s \frac{e(g^{r_1}, (V \prod_{i=1}^{k} H^{\mathrm{I_i}})^s Z_3) e(g^{r_2}, F^s Z_2)}{e(w(v \prod_{i=1}^{k} h_i^{\mathrm{I_i}})^{r_1} f^{r_2}, G^s Z_1)}$$

$$= \mathrm{Msg} \cdot e(g, w)^s \frac{e(g^{r_1}, (V \prod_{i=1}^{k} H^{\mathrm{I_i}})^s) e(g^{r_2}, f^s)}{e(w(v \prod_{i=1}^{k} h_i^{\mathrm{I_i}})^{r_1} f^{r_2}, g^s)} = \mathrm{Msg}$$

The second equality holds because $e(h_p, h_q) = 1$ for all $h_p \in \mathbb{G}_p$ and $h_q \in \mathbb{G}_q$.

3.2 Proof of Security

In this section, we explain the security of our construction. Our construction is similar to that of BBG-HIBE except for the blinding factors and the re-randomization subkey of the private key. Since the re-randomization subkey does not contain an element of the master key, w, adding the re-randomization subkey does not effect the semantic security. Therefore, we can demonstrate the semantic security of our construction in a similar manner to that for BBG-HIBE.

To prove anonymity, we use hybrid steps similar to that of [11]. The security of HVE is based on the composite 3-party Diffie-Hellman assumption, however we use the L-cDH which is a stronger assumption than c3DH, where L is the maximum hierarchy depth. The reason we introduce and use the L-cDH assumption is like that the semantic securities of our scheme and BBG-HIBE scheme are based on the Decision L-BDHI* assumption which is a stronger assumption than the Decision BDH assumption and depends on the maximum hierarchy depth L. The private key of BBG-HIBE has delegation key elements whose number depends on the maximum hierarchy depth. Therefore it may be that an adversary attacking the BBG-HIBE scheme or our scheme can get more information from the private key extraction queries than from other HIBE schemes in which the private key does not contain delegation key elements and is secure under the Decision BDH assumption, for example, GS-HIBE and BB-HIBE. The Decision L-BDHI assumption and the L-cDH assumption guarantee that any computationally bounded adversary can get no information about the message and the identity, respectively, from the chosen private keys and the challenge ciphertext with reasonable constraints.

Theorem 1. *If group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ_1)-Decision L-wBDHI* assumption and (t, ϵ_2)-L-cDH assumption, then our HIBE scheme with maximum hierarchy depth L is* $(q_s, \hat{t_1}, \hat{\epsilon_1})$-IND-sID-CPA *secure and* $(q_s, \hat{t_2}, \hat{\epsilon_2})$-ANON-sID-CPA *secure with $\hat{t_1}$, $\hat{t_2} = \Theta(t)$, $\hat{\epsilon_1} = \Theta(\epsilon_1 + \epsilon_2)$, and $\hat{\epsilon_2} = \Theta(\epsilon_1 + \epsilon_2/(1 - \frac{p+q-1}{n})^{q_s})$.*

We prove Theorem 1 using hybrid experiments under the Decision L-wBDHI* assumption and L-cDH assumption.

$\mathrm{Game}_1 : \mathrm{CT}_1 = [C_1,\ C_2,\ C_3,\ C_4]$
$\mathrm{Game}_2 : \mathrm{CT}_2 = [C_1 \cdot R_p,\ C_2,\ C_3,\ C_4]$
$\mathrm{Game}_3 : \mathrm{CT}_3 = [C_1 \cdot R = R_1,\ C_2,\ C_3,\ C_4]$

$\text{Game}_4 : \text{CT}_4 = [R_1,\ R_2,\ C_3,\ C_4]$
$\text{Game}_5 : \text{CT}_5 = [R_1,\ R_2,\ R_3,\ R_4]$

where R_p is a randomly chosen element from $\mathbb{G}_{T,p}$; R, R_1 are uniformly distributed in $\mathbb{G}_T$; and R_2, R_3, R_4 are uniformly distributed in $\mathbb{G}$.

We show that under the Decision L-wBDHI* assumption and L-cDH assumption, there are no algorithms that distinguish between Game$_1$ and Game$_2$, or between Game$_2$ and Game$_3$, or between Game$_3$ and Game$_4$, or between Game$_4$ and Game$_5$. Challenge ciphertext CT$_5$ is composed of four random group elements, so it does not leak any information about the message or the identity. Therefore indistinguishability between games prove Theorem 1. First, we prove the indistinguishability between games, and next we complete the proof of Theorem 1.

Indistinguishability between Game$_1$ and Game$_2$.

Lemma 1. *If group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ)-Decision L-wBDHI* assumption, there is no adversary with running time t that distinguishes between* Game$_1$ *and* Game$_2$ *with advantage ϵ.*

Proof. We assume that there exists adversary $\mathcal{A}$ that distinguishes between Game$_1$ and Game$_2$ with advantage ϵ. We show that there is a simulator $\mathcal{B}$ using $\mathcal{A}$ to solve the Decision L-wBDH* problem with advantage ϵ. The proof is similar to the proof of the semantic security of BBG-HIBE except for the treatment of the re-randomization key in our proof.

The challenger makes a challenge pair $(\mathbf{Z}, T')$ of the Decision L-wBDHI* assumption. which is defined in Section 2.3 and gives the challenge pair to simulator $\mathcal{B}$. Let $A_i = g_p^{a^i}$ where $g_p^{a^i}$ is defined in $\mathbf{Z}$ for $1 \le i \le L+1$.

Initialization. Adversary $\mathcal{A}$ chooses challenge identity $\text{ID} = [\text{I}_1, \text{I}_2, \cdots, \text{I}_\text{m}]$, and sends it to simulator $\mathcal{B}$. Then, $\mathcal{B}$ sets $\text{I}_{\text{m}+1} = \cdots = \text{I}_\text{L} = 0$. Hence, a simulator can always consider the length of the challenge identity as L.

Setup. $\mathcal{B}$ chooses random integers and random elements

$$\gamma, x, y, z, x_1, \cdots, x_L \in \mathbb{Z}_n, \qquad R_g, R_f, R_v, R_{h,1}, \cdots, R_{h,l} \in \mathbb{G}_q.$$

A random element of $\mathbb{G}_p(\mathbb{G}_q)$ can be chosen by raising $g_p(g_q,$ respectively) to random exponents from $\mathbb{Z}_n$. $\mathcal{B}$ sets $G = g_p R_g$, $F = g_p^z R_f$, $V = (g_p^y \cdot \prod_{i=1}^{L} (A_{L-i+1})^{\text{I}_i}) R_v$, $H_i = g_p^{x_i} / A_{L-i+1} R_{h,i}$ for $1 \le i \le L$, and $E = e(A_1, A_L \cdot g_p^{\gamma})$. Then, $\mathcal{B}$ publishes system parameters as

$$params \leftarrow [g_q, G, F, V, H_1, \cdots, H_L, E]$$

where *params* generated by $\mathcal{B}$ has the same distribution as that of an actual scheme. The master key w corresponding to the system parameters is $(A_L \cdot g_p^{\gamma})^a = A_{L+1} \cdot A_1^{\gamma}$. Since $\mathcal{B}$ does not have A_{L+1}, $\mathcal{B}$ does not know the master key.

Query Phase1. $\mathcal{A}$ queries the private key for $\text{ID}^* = [\text{I}_1^*, \text{I}_2^*, \cdots, \text{I}_\text{u}^*]$, where $u \le L$ is distinct from ID and all its prefixes. This private query is carried on an adaptively chosen identity by $\mathcal{A}$. Let k be the smallest integer such that $\text{I}_\text{k} \ne \text{I}_\text{k}^*$.

Then, first $\mathcal{B}$ generates the private key corresponding to $[\mathrm{I}_1^*, \mathrm{I}_2^*, \cdots, \mathrm{I}_k^*]$ and runs the *Derive* algorithm to make ID^*. $\mathcal{B}$ first chooses random integers $r_1, r_2 \in \mathbb{Z}_n$. We posit $\hat{r_1} = r_1 + \frac{a^k}{\mathrm{I}_k^* - \mathrm{I}_k}$. Next, it generates $\mathrm{Pvk}^{\mathrm{ID}^*} = [\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}, \mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}]$. We observe the first component of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$.

$$w(v\prod_{i=1}^{k} h_i^{\mathrm{I_i}^*})^{\hat{r_1}} f^{r_2} = w \cdot (v\prod_{i=1}^{k} h_i^{\mathrm{I_i}^*})^{r_1} f^{r_2} \cdot (v\prod_{i=1}^{k} h_i^{\mathrm{I_i}^*})^{\frac{a^k}{\mathrm{I}_k^* - \mathrm{I}_k}}$$

Since v is $g_p^y \cdot \prod_{i=1}^{L} (A_{L-i+1})^{\mathrm{I_i}}$ and h_i is $g_p^{x_i}/A_{L-i+1}$ and f is g_p^z which can be obtained by removing the blinding factor from V, H_i and F, respectively, and r_1, r_2 are chosen by simulator, $\mathcal{B}$ can compute the second term in the above expression. We focus on the product of the first and third terms in the above expression, $w \cdot (v\prod_{i=1}^{k} h_i^{\mathrm{I_i}})^{\frac{a^k}{\mathrm{I}_k^* - \mathrm{I}_k}}$. Then

$$\begin{aligned}
w \cdot (v\prod_{i=1}^{k} h_i^{\mathrm{I_i}^*})^{\frac{a^k}{\mathrm{I}_k^* - \mathrm{I}_k}} &= A_{L+1}A_1^{\gamma} \cdot (g_p^y \prod_{i=1}^{L} (A_{L-i+1})^{\mathrm{I_i}} \prod_{i=1}^{k} (g_p^{x_i}/A_{L-i+1})^{\mathrm{I_i}^*})^{\frac{a^k}{\mathrm{I}_k^* - \mathrm{I}_k}} \\
&= A_{L+1}A_1^{\gamma} \cdot (g_p^y A_{L-k+1}^{\mathrm{I}_k - \mathrm{I}_k^*} \prod_{i=k+1}^{L} (A_{L-i+1})^{\mathrm{I_i}} \prod_{i=1}^{k} g_p^{x_i \mathrm{I}_i^*})^{\frac{a^k}{\mathrm{I}_k^* - \mathrm{I}_k}} \\
&= A_{L+1}A_1^{\gamma} \cdot (A_k^y A_{L+1}^{\mathrm{I}_k - \mathrm{I}_k^*} \prod_{i=k+1}^{L} (A_{L+k-i+1})^{\mathrm{I_i}} \prod_{i=1}^{k} A_k^{x_i \mathrm{I}_i^*})^{\frac{1}{\mathrm{I}_k^* - \mathrm{I}_k}} \\
&= A_1^{\gamma} \cdot (A_k^y \prod_{i=k+1}^{L} (A_{L+k-i+1})^{\mathrm{I_i}} \prod_{i=1}^{k} A_k^{x_i \mathrm{I}_i^*})^{\frac{1}{\mathrm{I}_k^* - \mathrm{I}_k}}
\end{aligned}$$

Since $\mathcal{B}$ knows all the terms in the above expression, it can compute the first component of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$. Since the remaining elements in $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$ do not involve A_{L+1}, $\mathcal{B}$ can compute all of them. $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$ is distributed as if $\hat{r_1} = r_1 + \frac{a^k}{\mathrm{I}_k - \mathrm{I}_k^*}$ and r_2 are the randomness of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$. Since $\hat{r_1}$ and r_2 are uniformly and independently distributed in $\mathbb{Z}_n$, $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}*}$ has the same distribution as that of the actual key distribution.

To generate $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$, $\mathcal{B}$ choose $s_1, s_2, t_1, t_2 \in \mathbb{Z}_n$. Since no elements in $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ associate with master key w, $\mathcal{B}$ can compute $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ using s_1, s_2, t_1, t_2 as its randomness. Since the random integers used in $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ have to satisfy equation $s_1 \cdot t_2 - s_2 \cdot t_1 \not\equiv 0 \mod p$ and $\not\equiv 0 \mod q$, the simulator has to check equation $g_p^{s_1 \cdot t_2 - s_2 \cdot t_1} \not\equiv 1$ and $g_q^{s_1 \cdot t_2 - s_2 \cdot t_1} \not\equiv 1$. If the random integers used in $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ do not satisfy the equation, then the simulator chooses other random integers s_1, s_2, t_1 and t_2 and repeats the same procedure until the equation does hold. Since the equation holds without probability $\frac{p+q-1}{n}$, this iteration will finish immediately. Therefore $\mathrm{Pvk}^{\mathrm{ID}^*}$ has the same distribution as that of the actual key distribution.

Challenge. $\mathcal{A}$ sends a message Msg $\in \mathbb{G}$ to $\mathcal{B}$. Then, $\mathcal{B}$ picks random elements Z_1, Z_2 and Z_3 from $\mathbb{G}_q$ and outputs a challenge ciphertext

$$\mathrm{CT} = [\mathrm{Msg} \cdot T' \cdot e(A_1, h^{\gamma}), \quad h \cdot Z_1, \quad h^z \cdot Z_2, \quad h^{y+\Sigma_{i=1}^{L} \mathrm{I}_i x_i} \cdot Z_3],$$

where h and T' are given from challenge pair $(\mathbf{Z}, T')$ We consider h as g_p^c for some unknown $c \in \mathbb{Z}_n$.

If $T' = T$, then CT is equal to

$$[\mathrm{Msg}{\cdot}e(g_p, g_p^c)^{a^{L+1}} e(g_p^a, g_p^{c\gamma}), g_p^c Z_1, (g_p^z)^c Z_2, (\prod_{i=1}^{L} (g_p^{x_i}/A_{L-i+1})^{\mathrm{I}_i} g_p^y \prod_{i=1}^{L} A_{L-i+1}^{\mathrm{I}_i})^c Z_3]$$

$$= [\mathrm{Msg} \cdot e(A_1, A_L g_p^{\gamma})^c, \quad G^c Z_1', \quad F^c Z_2', \quad (V \prod_{i=1}^{L} H_i^{\mathrm{I}_i})^c \ Z_3']$$

$$= [\mathrm{Msg} \cdot e(A_1, A_L g_p^{\gamma})^c, \quad G^c Z_1', \quad F^c Z_2', \quad (V \prod_{i=1}^{m} H_i^{\mathrm{I}_i})^c Z_3'].$$

Therefore, CT is a ciphertext of Game$_1$. Otherwise, T is a uniformly and independently chosen element from $\mathbb{G}_T$. In that case, the first component of ciphertext is random from the adversarial point of view. Therefore, CT is a ciphertext of Game$_2$.

Query Phase2. $\mathcal{A}$ adaptively queries $\mathcal{B}$ with the same constraints as in Query Phase 1. $\mathcal{B}$ sends corresponding private keys as before.

Guess. $\mathcal{B}$ outputs the same bit as $\mathcal{A}$; i.e., if $\mathcal{A}$ outputs 1 (Game$_1$), then $\mathcal{B}$ also outputs 1 ($T' = T$). Since $\mathcal{B}$ played Game$_1$ with $T' = T$ and played Game$_2$ with T' as a random element from $\mathbb{G}_p$, $\mathcal{B}$'s advantage in the L-wBDHI* game is exactly ϵ, the same as $\mathcal{A}$'s. □

Indistinguishability between Game$_2$ and Game$_3$.

Lemma 2. *If group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ)-BSD assumption, there is no adversary with running time t that distinguishes between* Game$_2$ *and* Game$_3$ *with advantage ϵ.*

Proof. We assume that there exists adversary $\mathcal{A}$ distinguishing between Game$_2$ and Game$_3$ with ϵ advantage. We show that there is a simulator $\mathcal{B}$ using $\mathcal{A}$ to solve the BSD problem with advantage ϵ.

The challenger makes a challenge pair ($\mathbf{Z}$, T') of the BSD problem, which is defined in Section 2.3 and give the challenge pair to simulator $\mathcal{B}$.

Initialization. $\mathcal{A}$ sends a challenge identity ID to $\mathcal{B}$.
Setup. $\mathcal{B}$ generates system parameters as an actual setup algorithm. $\mathcal{B}$ can choose all random elements from $\mathbb{G}_p$ and $\mathbb{G}_p$ by using g_p and g_q.

Query Phase1. $\mathcal{A}$ queries $\mathcal{B}$ and $\mathcal{B}$ responds to queries as the actual key generation center.

Challenge. $\mathcal{A}$ sends message Msg, to $\mathcal{B}$. $\mathcal{B}$ outputs a normal ciphertext with the exception that the its first component is multiplied by T'.

Query Phase2. $\mathcal{A}$ adaptively queries $\mathcal{B}$ with the same constraints as in Query Phase 1. $\mathcal{B}$ sends corresponding private keys as before.

Guess. $\mathcal{B}$ outputs the same bit as $\mathcal{A}$; i.e., if $\mathcal{A}$ outputs 1 (Game$_2$), then $\mathcal{B}$ also outputs 1 ($T' = T$). Since $\mathcal{B}$ played Game$_2$ with T' is a random element from $\mathbb{G}_p$ and played Game$_3$ with T' as a random element from $\mathbb{G}$, $\mathcal{B}$'s advantage in the BSD game is exactly ϵ, the same as $\mathcal{A}$'s. $\square$

Indistinguishability between Game$_3$ and Game$_4$

Lemma 3. *If group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ)-L-cDH assumption, there is no adversary with running time t that makes at most q_s key extraction queries and distinguishes between* Game$_3$ *and* Game$_4$ *with advantage* $\epsilon/(1 - \frac{p+q-1}{n})^{q_s}$.

Proof. We assume that there exists an adversary $\mathcal{A}$ distinguishing between Game$_3$ and Game$_4$ with advantage ϵ'. We show that there is a simulator $\mathcal{B}$ using $\mathcal{A}$ to solve the L-cDH problem with advantage $\epsilon' \cdot (1 - \frac{p+q-1}{n})^{q_s}$.

The challenger makes a challenge pair ($\mathbf{Z}$, T') of L-cDH which is defined in Section 2.3 and gives the challenge pair to simulator $\mathcal{B}$. Let $A_i = g_p^{a^i}$, $B = A_{L+1}R'_1$ and $C = A^b_{L+1}R'_2$ where $g_p^{a^i}$, R'_1 and R'_2 are defined in $\mathbf{Z}$ for $0 \leq i \leq L + 1$.

Initialization. $\mathcal{A}$ chooses a challenge $\mathrm{ID} = [\mathrm{I}_1, \mathrm{I}_2, \cdots, \mathrm{I}_\mathrm{m}]$ and sends it to simulator $\mathcal{B}$.

Setup. $\mathcal{B}$ chooses random integers and random elements

$$x, y, z, x_1, \cdots, x_L \in \mathbb{Z}_n, \qquad w \in \mathbb{G}_p, \qquad R_g, R_f, R_v, R_{h,1}, \cdots, R_{h,l} \in \mathbb{G}_q.$$

$\mathcal{B}$ puts $G = B^x R_g$, $F = g_p^z R_f$, $H_i = A_i^{x_i} R_{h,i}$ for $1 \leq i \leq L$, $V = (g_p^y / \prod_{j=1}^m H_j^{I_j})$ R_v, and $E = e(B^x, w)$. Then, $\mathcal{B}$ publishes system parameters as

$$params \leftarrow [g_q, G, F, V, H_1, \cdots, H_L, E]$$

Query Phase1. $\mathcal{A}$ queries the private key for $\mathrm{ID}^* = [\mathrm{I}_1^*, \mathrm{I}_2^*, \cdots, \mathrm{I}_\mathrm{u}^*]$ where $u \leq L$ distinct from ID and all its prefixes. This private query of an adaptively chosen identity is carried out by $\mathcal{A}$. Let k be the smallest integer such that $\mathrm{I_k} \neq \mathrm{I}_\mathrm{k}^*$. Then, first $\mathcal{B}$ generates the private key corresponding to $[\mathrm{I}_1^*, \mathrm{I}_2^*, \cdots, \mathrm{I}_\mathrm{k}^*]$ and runs *Derive* algorithm to make ID^*. $\mathcal{B}$ first choose random integers $r_1, r_2 \in \mathbb{Z}_n$. We posit $\hat{r_1} = \frac{z}{a^k} r_1 + \frac{z}{a^{k+1}} r_2, \hat{r_2} = -\frac{y}{a^k} r_1 - (\frac{x_k(\mathrm{I}_\mathrm{k}^* - \mathrm{I_k})}{a} + \frac{y}{a^{k+1}}) r_2$. Next, the algorithm generates $\mathrm{Pvk}^{\mathrm{ID}^*} = [\mathrm{Pvk}_\mathrm{d}^{\mathrm{ID}^*}, \mathrm{Pvk}_\mathrm{r}^{\mathrm{ID}^*}]$. We observe the first component of $\mathrm{Pvk}_\mathrm{d}^{\mathrm{ID}^*}$, $w(v \prod_{i=1}^k h_i^{\mathrm{I_i}^*})^{\hat{r_1}} f^{\hat{r_2}}$. Since we know that v, h_i, f are the same as elements by removing blinding factors from V, H_i, F, respectively, we can rewrite the above component as follows:

$$w(v\prod_{i=1}^{k} h_i^{\mathrm{I_i}^*})^{\hat{r_1}} f^{\hat{r_2}} = w((g_p^y/\prod_{j=1}^{m} A_j^{x_j I_j})\prod_{i=1}^{k} A_i^{x_i \mathrm{I}_\mathrm{i}^*})^{\hat{r_1}} g_p^{z\hat{r_2}}$$

We focus on the exponent of g_p in $((g_p^y/\prod_{j=1}^{m} A_j^{x_j I_j})\prod_{i=1}^{k} A_i^{x_i \mathrm{I}_\mathrm{i}^*})^{\hat{r_1}} g_p^{z\hat{r_2}}$. It is

$$(y - \sum_{j=1}^{m} a^j x_j I_j + \sum_{i=1}^{k} a^i x_i I_i^*)\hat{r_1} + z\hat{r_2}$$

$$= (y - \sum_{j=k+1}^{m} a^j x_j I_j + a^k x_k (I_k^* - I_k))\hat{r_1} + z\hat{r_2}$$

$$= (y - \sum_{j=k+1}^{m} a^j x_j I_j + a^k x_k (I_k^* - I_k))(\frac{z}{a^k} r_1 + \frac{z}{a^{k+1}} r_2) + z(-\frac{y}{a^k} r_1 - (\frac{x_k(\mathrm{I}_\mathrm{k}^* - \mathrm{I}_\mathrm{k})}{a} + \frac{y}{a^{k+1}}) r_2)$$

$$= (-\sum_{j=k+1}^{m} a^{j-k} x_j I_j z + x_k (I_k^* - I_k) z) r_1 - \sum_{i=k+1}^{m} a^{i-k-1} x_i I_i z r_2$$

Since the exponent involves $a^1, \cdots, a^{m-k}$, x_j, z, r_1, r_2, ID and ID^*, $\mathcal{B}$ can compute the first component of $\mathrm{Pvk_d^{ID^*}}$. The remaining elements in $\mathrm{Pvk_d^{ID^*}}$ are $g^{\hat{r_1}}$, $g^{\hat{r_2}}$ and $h_i^{\hat{r_1}}$ for $k+1 \le i \le L$. Note that g and h_i are A_{L+1}^x and $A_i^{x_i}$, respectively, which are elements with blinding factors removed from G and H_i, respectively. Since the second component $g^{\hat{r_1}}$ is equal to $A_{L+1}^{x\hat{r_1}} = A_{L+1}^{x(\frac{z}{a^k} r_1 + \frac{z}{a^{k+1}} r_2)} = A_{L-k+1}^{xzr_1} A_{L-k}^{xzr_2}$, $\mathcal{B}$ can compute the second component of $\mathrm{Pvk_d^{ID^*}}$. Similarly, $\mathcal{B}$ can compute all the remaining elements of $\mathrm{Pvk_d^{ID^*}}$. Since $\hat{r_1}$ and $\hat{r_2}$ are uniformly and independently distributed in $\mathbb{Z}_n$, $\mathrm{Pvk_d^{ID*}}$ has the same distribution as an actual key distribution.

Next, $\mathcal{B}$ generates $\mathrm{Pvk_r^{ID^*}}$. Every component in $\mathrm{Pvk_r^{ID^*}}$ is the same as in $\mathrm{Pvk_d^{ID^*}}$ except for w of the first and $(L-k+4)$th components and for using different randomness. Since the procedure for generating $\mathrm{Pvk_d^{ID^*}}$ will work without w, $\mathcal{B}$ can generate $\mathrm{Pvk_r^{ID^*}}$ in a similar manner to generating $\mathrm{Pvk_d^{ID^*}}$. The details of this are highly similar to those of $\mathrm{Pvk_r^{ID^*}}$, so they are omitted. We let $s_1, s_2, t_1, t_2 \in \mathbb{Z}_n$ be random integers used for generating $\mathrm{Pvk_r^{ID^*}}$ and let $\hat{s_1} = \frac{z}{a^k} s_1 + \frac{z}{a^{k+1}} s_2$, $\hat{s_2} = -\frac{y}{a^k} s_1 - (\frac{x_k(\mathrm{I}_\mathrm{k}^* - \mathrm{I}_\mathrm{k})}{a} + \frac{y}{a^{k+1}}) s_2$, $\hat{t_1} = \frac{z}{a^k} t_1 + \frac{z}{a^{k+1}} t_2$, $\hat{t_2} = -\frac{y}{a^k} t_1 - (\frac{x_k(\mathrm{I}_\mathrm{k}^* - \mathrm{I}_\mathrm{k})}{a} + \frac{y}{a^{k+1}}) t_2$. Then $\mathrm{Pvk_r^{ID^*}}$ is distributed as if $\hat{s_1}, \hat{s_2}, \hat{t_1}$, and $\hat{t_2}$ are the randomness of $\mathrm{Pvk_r^{ID^*}}$. Since $\hat{s_1}$, $\hat{s_2}$, $\hat{t_1}$, and $\hat{t_2}$ are uniformly and independently distributed in $\mathbb{Z}_n$, four integers satisfy equations $\hat{s_1} \cdot \hat{t_2} - \hat{s_2} \cdot \hat{t_1} \not\equiv 0 \mod p$ and $\not\equiv 0 \mod q$ with probability $1 - \frac{p+q-1}{n}$. Therefore, the private key $\mathrm{Pvk^{ID^*}}$ generated by the simulator has the same structure and distribution as that of actual private key with probability $1 - \frac{p+q-1}{n}$.

Challenge. $\mathcal{A}$ sends message Msg to $\mathcal{B}$. $\mathcal{B}$ discards Msg and selects random elements $R \in \mathbb{G}$ and $Z_1, Z_2, Z_3 \in \mathbb{G}_q$. $\mathcal{B}$ sends ciphertext CT=$[R, C^x Z_1, T'^z Z_2, T'^y Z_3]$.

If $T' = T$, then CT is equal to

$$[R, (A_{L+1}^b R_2')^x Z_1, (g_p^b R_3')^z Z_2, (g_p^b R_3')^y Z_3] = [R, G^b Z_1', F^b Z_2', (\prod_{i=1}^{m} H_i^{\mathrm{I_i}} V)^b Z_3'].$$

Therefore CT is a ciphertext of Game$_3$. Otherwise, T can be written by $g_p^r R_3''$ as an element from $\mathbb{G}$ where r is a random integer chosen from $\mathbb{Z}_n$ and R_3'' is a random element chosen from $\mathbb{G}_q$. Then, CT is equal to

$$[R, (A_{L+1}^b R_2')^x Z_1, (g_p^r R_3'')^z Z_2, (g_p^r R_3'')^y Z_3] = [R, G^b Z_1', F^r Z_2', (V \prod_{i=1}^{m} H_i^{\mathrm{I_i}})^r Z_3'].$$

From an adversarial viewpoint, b and r first appear in ciphertext CT and both are integers uniformly and independently chosen from $\mathbb{Z}_n$. The third and fourth components in CT share the same random r. However, the second component uses random b independent from r. Therefore the second component of CT is a random element from the adversarial viewpoint, and CT is a ciphertext of Game$_4$.

Query Phase2. $\mathcal{A}$ adaptively queries $\mathcal{B}$ with the same constraints in Query Phase 1. $\mathcal{B}$ sends corresponding private keys as before.

Guess. $\mathcal{B}$ outputs the same bit as $\mathcal{A}$, i.e., if $\mathcal{A}$ outputs 1 (Game$_3$), then $\mathcal{B}$ also outputs 1 ($T' = T$). If $\mathcal{A}$ queried q_s times in total in *Query Phase1* and *Query Phase2*, then $\mathcal{B}$ responded with corresponding private keys having the same distribution as that of actual keys with probability $(1 - \frac{p+q-1}{n})^{q_s}$. Therefore $\mathcal{B}$'s advantage in the L-cDH game is $\epsilon' \cdot (1 - \frac{p+q-1}{n})^{q_s}$. □

Indistinguishability between Game$_4$ and Game$_5$

Lemma 4. *If group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ)-L-cDH assumption, there is no adversary with running time t that makes at most q_s key extraction queries and distinguishes between* Game$_4$ *and* Game$_5$ *with advantage* $\epsilon/(1 - \frac{p+q-1}{n})^{q_s}$.

Proof. We assume that there exists adversary $\mathcal{A}$ distinguishing between Game$_4$ and Game$_5$ with non-negligible advantage ϵ'. We show that there is a simulator $\mathcal{B}$ using $\mathcal{A}$ to solve the L-cDH problem with advantage $\epsilon' \cdot (1 - \frac{p+q-1}{n})^{q_s}$.

The challenger makes a challenge pair ($\mathbf{Z}$, T') of L-cDH which is defined in Section 2.3 and give the challenge pair to simulator $\mathcal{B}$. Let $A_i = g_p^{a^i}$, $B = A_{L+1} R_1'$ and $C = A_{L+1}^b R_2'$ where $g_p^{a^i}$, R_1' and R_2' are defined in $\mathbf{Z}$ for $0 \le i \le L+1$.

Initialization. $\mathcal{A}$ chooses a challenge ID $= [\mathrm{I_1}, \mathrm{I_2}, \cdots, \mathrm{I_m}]$ and sends it to simulator $\mathcal{B}$.

Setup. $\mathcal{B}$ chooses random integers and random elements

$$x, y, z, x_1, \cdots, x_L \in \mathbb{Z}_n, \qquad w \in \mathbb{G}_p, R_g, R_f, R_v, R_{h,1}, \cdots, R_{h,l} \in \mathbb{G}_q.$$

$\mathcal{B}$ puts $G = g_p^x R_g$, $F = B^z R_f$, $V = (g_p^y / \prod_{j=1}^m H_j^{I_j}) R_v$, $H_i = A_{L+1-i}^{x_i} R_{h,i}$ for $1 \leq i \leq L$, and $E = e(g_p^x, w)$. Then, $\mathcal{B}$ publishes system parameters as

$$\mathit{params} \leftarrow [g_q, G, F, V, H_1, \cdots, H_L, E].$$

Query Phase1. $\mathcal{A}$ queries the private key for $\mathrm{ID}^* = [\mathrm{I}_1^*, \mathrm{I}_2^*, \cdots, \mathrm{I}_u^*]$ where $u < L$ is distinct from ID and all its prefixes. This private query is carried out on an identity adaptively chosen by $\mathcal{A}$. Let k be the smallest integer such that $\mathrm{I}_k \neq \mathrm{I}_k^*$. Then, first $\mathcal{B}$ generates the private key corresponding to $[\mathrm{I}_1^*, \mathrm{I}_2^*, \cdots, \mathrm{I}_k^*]$ and runs the *Derive* algorithm to make ID^*. $\mathcal{B}$ first chooses random integers $r_1, r_2 \in \mathbb{Z}_n$. We posit $\hat{r_1} = a^k z r_1 + r_2, \hat{r_2} = -x_k(\mathrm{I}_k^* - \mathrm{I}_k) r_1$. Next, the algorithm generates $\mathrm{Pvk}^{\mathrm{ID}^*} = [\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}, \mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}]$. We observe the first component in $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$, $w(v \prod_{i=1}^k h_i^{\mathrm{I_i}^*})^{\hat{r_1}} f^{\hat{r_2}}$. Since we know that v, h_i, f are same as elements by removing blinding factor from V, H_i, F, respectively, we can rewrite above component as follows:

$$w(v \prod_{i=1}^k h_i^{\mathrm{I_i}^*})^{\hat{r_1}} f^{\hat{r_2}} = w((g_p^y / \prod_{j=1}^m A_{L+1-j}^{x_j I_j}) \prod_{i=1}^k A_{L+1-i}^{x_i \mathrm{I}_i^*})^{\hat{r_1}} A_{L+1}^{z \hat{r_2}}$$

We focus on the exponent of g_p in $((g_p^y / \prod_{j=1}^m A_{L+1-j}^{x_j I_j}) \prod_{i=1}^k A_{L+1-i}^{x_i \mathrm{I}_i^*})^{\hat{r_1}} A_{L+1}^{z \hat{r_2}}$. It is

$$(y - \sum_{j=1}^m a^{L+1-j} x_j I_j + \sum_{i=1}^k a^{L+1-i} x_i I_i^*) \hat{r_1} + a^{L+1} z \hat{r_2}$$

$$= (y - \sum_{j=k+1}^m a^{L+1-j} x_j I_j + a^{L+1-k} x_k (I_k^* - I_k)) \hat{r_1} + a^{L+1} z \hat{r_2}$$

$$= (y - \sum_{j=k+1}^m a^{L+1-j} x_j I_j + a^{L+1-k} x_k (I_k^* - I_k))(a^k z r_1 + r_2) + a^{L+1} z (-x_k(\mathrm{I}_k^* - \mathrm{I}_k) r_1)$$

$$= (y a^k z - \sum_{j=k+1}^m a^{L+k+1-j} x_j I_j z) r_1 + (y - \sum_{j=k+1}^m a^{L+1-j} x_j I_j + a^{L+1-k} x_k (I_k^* - I_k)) r_2$$

Since the exponent involves $a^1, \cdots, a^L$, x_j, z, r_1, r_2 and identities, $\mathcal{B}$ can compute the first component of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$. The remaining elements in $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$ are $g^{\hat{r_1}}$, $g^{\hat{r_2}}$ and $h_i^{\hat{r_1}}$ for $k+1 \leq i \leq L$. Note that g and h_i are g_p^x and $A_{L+1-i}^{x_i}$, respectively, which are elements with blinding factors removed from G and H_i, respectively. Since the second component $g^{\hat{r_1}}$ is equal to $g_p^{x \hat{r_1}} = A_k^{x z r_1} g_p^{x r_2}$, $\mathcal{B}$ can compute the second component of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$. Similarly, $\mathcal{B}$ can compute all the remaining elements of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$. Since $\hat{r_1}$ and $\hat{r_2}$ are uniformly and independently distributed in $\mathbb{Z}_n$, $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}*}$ has the same distribution as an actual key distribution.

Next, $\mathcal{B}$ generates $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$. Every component in $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ is the same as in $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$ except for w of the first and $(L-k+4)$th components and for using different randomness. Since generating procedure of $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$ will work without

w, $\mathcal{B}$ can generate $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ is a similar manner to generating $\mathrm{Pvk}_{\mathrm{d}}^{\mathrm{ID}^*}$. The details of this are highly similar to those of $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$, so they are omitted. We let $s_1, s_2, t_1, t_2 \in \mathbb{Z}_n$ be random integers used for generating $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ and let

$$\hat{s_1} = a^k z s_1 + s_2,\;\; \hat{s_2} = -x_k(\mathrm{I}_{\mathrm{k}}^* - \mathrm{I}_{\mathrm{k}})s_1,\;\; \hat{t_1} = a^k z t_1 + t_2, \text{ and } \hat{t_2} = -x_k(\mathrm{I}_{\mathrm{k}}^* - \mathrm{I}_{\mathrm{k}})t_1.$$

Then $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$ is distributed as if $\hat{s_1}, \hat{s_2}, \hat{t_1}$, and $\hat{t_2}$ is the randomness of $\mathrm{Pvk}_{\mathrm{r}}^{\mathrm{ID}^*}$. Since $\hat{s_1}$, $\hat{s_2}$, $\hat{t_1}$, and $\hat{t_2}$ are uniformly and independently distributed in $\mathbb{Z}_n$, four integers satisfy equations $\hat{s_1} \cdot \hat{t_2} - \hat{s_2} \cdot \hat{t_1} \not\equiv 0 \mod p$ and $\not\equiv 0 \mod q$ with probability $1 - \frac{p+q-1}{n}$. Therefore, the private key $\mathrm{Pvk}^{\mathrm{ID}^*}$ generated by the simulator has the same structure and distribution as that of actual private key with probability $1 - \frac{p+q-1}{n}$.

Challenge. $\mathcal{A}$ sends message Msg to $\mathcal{B}$. $\mathcal{B}$ discards Msg and selects random elements $R, R' \in \mathbb{G}$, $Z_1, Z_2 \in \mathbb{G}_q$. $\mathcal{B}$ sends ciphertext $\mathrm{CT} = [R, R', C^z Z_1, T'^y Z_2]$.
If $T' = T$, then

$$\mathrm{CT} = [R, R', (A_{L+1}^b R_2')^z Z_1, (g_p^b R_3')^y Z_2] = [R, R', F^b Z_1', (V \prod_{i=1}^{m} H_i^{\mathrm{I}_{\mathrm{i}}})^b Z_2'].$$

Therefore, CT is a ciphertext of Game_4. Otherwise, T can be written by $g_p^r R_3''$ as an element from $\mathbb{G}$, where r is a random integer chosen from $\mathbb{Z}_n$, and R_3'' is a random element chosen from $\mathbb{G}_q$. Then,

$$\mathrm{CT} = [R, R', (A_{L+1}^b R_2')^z Z_1, (g_p^r R_3'')^y Z_2] = [R, R', F^b Z_1', (V \prod_{i=1}^{m} H_i^{\mathrm{I}_{\mathrm{i}}})^r Z_2'].$$

From an adversarial viewpoint, b and r first appear in ciphertext CT and both are integers uniformly and independently chosen from $\mathbb{Z}_n$. Therefore, the third and fourth components of CT are independent random elements from the adversarial viewpoint, and CT is a ciphertext of Game_5.

Query Phase2. $\mathcal{A}$ adaptively queries $\mathcal{B}$ with the same constraints as in Query Phase 1. $\mathcal{B}$ sends corresponding private keys as before.

Guess. $\mathcal{B}$ outputs the same bit as $\mathcal{A}$, i.e., if $\mathcal{A}$ outputs 1 (Game_4), then $\mathcal{B}$ also outputs 1 ($T' = T$). If $\mathcal{A}$ queried q_s times in total in *Query Phase1* and *Query Phase2*, then $\mathcal{B}$ responded with corresponding private keys having the same distribution as that of actual keys with probability $(1 - \frac{p+q-1}{n})^{q_s}$. Therefore $\mathcal{B}$'s advantage in the L-cDH game is $\epsilon' \cdot (1 - \frac{p+q-1}{n})^{q_s}$. □

Proof of Theorem 1. If group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ_1)-Decision L-wBDHI* assumption and the (t, ϵ_2)-L-cDH assumption, then Lemma 1 and 2 show that there is no adversary with running time $\Theta(t)$ that makes at most q_s key extraction queries and distinguishes Game_1 and Game_3 with advantage $\epsilon_1 + \epsilon_2$. CT_3 does not leak any information about the message since there is no element involve in the message in CT_3. Therefore, if group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ_1)-Decision L-wBDHI* assumption and the (t, ϵ_2)-L-cDH assumption, our proposed HIBE scheme is $(q_s, \hat{t_1}, \hat{\epsilon_1})$-IND-sID-CPA secure with $\hat{t_1} = \Theta(t)$, $\hat{\epsilon_1} = \epsilon_1 + \epsilon_2$.

If group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ_2)-L-cDH assumption, then Lemma 3 and 4 show that there is no adversary with running time $\Theta(t)$ that makes at most q_s key extraction queries and distinguishes Game$_3$ and Game$_5$ with advantage $2\epsilon_2/(1 - \frac{p+q-1}{n})^{q_s}$.

CT$_5$ does not leak any information about the identity since all components of CT$_5$ are random group elements. Therefore, if group generator algorithm $\mathcal{G}$ satisfies the (t, ϵ_1)-Decision L-wBDHI* assumption and the (t, ϵ_2)-L-cDH assumption, then our proposed HIBE scheme is $(q_s, \hat{t_2}, \hat{\epsilon_2})$-ANON-sID-CPA secure with $\hat{t_2} = \Theta(t)$ and $\hat{\epsilon_2} = \epsilon_1 + \epsilon_2 + 2\epsilon_2/(1 - \frac{p+q-1}{n})^{q_s}$. This completes the proof. □

4 Comparison

The parameters of previous HIBE schemes, anonymous HIBE schemes and our proposed scheme are compared in table 1.

Table 1. HIBE schemes

	anonymity	# of group elements in public parameter	# of group elements in private key	# of group elements in ciphertext	# of pairing in decryption	security
GS-HIBE [17]	Non	2	k	$k+1$	k	w RO, f-ID
BB-HIBE [2]	Non	$L+3$	$k+1$	$k+2$	$k+1$	w/o RO, s-ID
BBG-HIBE [3]	Non	$L+3$	$L-k+2$	3	2	w/o RO, s-ID
BW-HIBE [12]	Ano	L^2+5L+7	$3L^2+(14-k)L$ $-3k+15$	$2L+6$	$2(L+2)$	w/o RO, s-ID
SW-dHVE [22]	Ano	$2L+6$	$(L-k)(k+5)$ $+k+3$	$L+4$	k	w/o RO, s-ID composit
This paper	Ano	$L+4$	$3(L-k+3)$	4	3	w/o RO, s-ID composit

L : the maximum depth of hierarchy, k : a depth of a corresponding identity,
Non: non-anonymous ID, Ano: anonymous ID
w/ RO : with random oracle, w/o RO : without random oracle
f-ID : full-ID, s-ID : selective-ID

Our scheme is the first reported constant size ciphertext anonymous HIBE scheme. Moreover, the computational cost is achieved the cheapest computational cost because during the decryption phase the computational cost is constant. The security proof can be shown without random oracles and our scheme achieves selective-ID CPA security.

5 Conclusion

We proposed an efficient anonymous Hierarchical Identity-Based Encryption scheme. The ciphertext of our scheme is only four group elements without depending on the depth of the hierarchy. Moreover, the computational cost of decryption is also efficient, just three bilinear pairings without depending on the hierarchy depth. The number of group elements in the public parameter as well as the private key of our anonymous HIBE are also the smallest among existing anonymous HIBE schemes.

The security of our scheme for a hierarchy depth L is selective-ID secure against a CPA adversary that was shown under the Decision L-wBDHI* assumption and the new L-composite Diffie-Hellman assumption without using random oracles. CCA2 security can be achieved by using techniques that are method of transforming from CPA-secure HIBE to CCA-secure HIBE, for example [14,8,9].

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 extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)
2. Boneh, D., Boyen, X.: Efficient selective-ID identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertexts. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
4. Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertexts. Cryptology ePrint Archive: Report 2005/015 (2005), http://eprint.iacr.org/2005/015
5. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public-key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
6. 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)
7. Boneh, D., Goh, E., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
8. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
9. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM Conference on Computer and Communications Security-CCS 2005. ACM Press, New York (2005)
10. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
11. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
12. 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)
13. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656. Springer, Heidelberg (2003)

14. Canetti, R., 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)
15. Chatterjee, S., Sarkar, P.: HIBE withe short public parameters without random oracle. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 145–160. Springer, Heidelberg (2006)
16. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
17. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
18. Horwitz, J., Lynn, B.: Towards hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
19. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
20. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. Cryptology ePrint Archive: Report 2007/404 (2007), http://eprint.iacr.org/2007/404
21. Mitsunari, S., Sakai, R., Kasahara, M.: A new traitor tracing. IEICE Transactions Fundamentals E85-A(2), 481–484 (2002)
22. Shi, E., Waters, B.: Delegating capabilities in predicate encryption systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 560–578. Springer, Heidelberg (2008)
23. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)