Hierarchical Identity Based Encryption with Constant Size Ciphertext*

Dan Boneh[1]**, Xavier Boyen[2], and Eu-Jin Goh[1]**

[1] Computer Science Department, Stanford University
{dabo,eujin}@cs.stanford.edu
[2] Voltage Inc., Palo Alto
xb@boyen.org

Abstract. We present a Hierarchical Identity Based Encryption (HIBE) system where the ciphertext consists of just three group elements and decryption requires only two bilinear map computations, *regardless of the hierarchy depth*. Encryption is as efficient as in other HIBE systems. We prove that the scheme is selective-ID secure in the standard model and fully secure in the random oracle model. Our system has a number of applications: it gives very efficient forward secure public key and identity based cryptosystems (with short ciphertexts), it converts the NNL broadcast encryption system into an efficient public key broadcast system, and it provides an efficient mechanism for encrypting to the future. The system also supports limited delegation where users can be given restricted private keys that only allow delegation to bounded depth. The HIBE system can be modified to support sublinear size private keys at the cost of some ciphertext expansion.

1 Introduction

An Identity Based Encryption (IBE) system [24, 5] is a public key system where the public key can be an arbitrary string such as an email address. A central authority uses a master key to issue private keys to identities that request them. Hierarchical IBE (HIBE) [17, 14] is a generalization of IBE that mirrors an organizational hierarchy. An identity at level k of the hierarchy tree can issue private keys to its descendant identities, but cannot decrypt messages intended for other identities (details are given in Section 2.1). The first construction for HIBE is due to Gentry and Silverberg [14] where security is based on the Bilinear Diffie-Hellman (BDH) assumption in the random oracle model. A subsequent construction due to Boneh and Boyen [1] gives an efficient (selective-ID secure) HIBE based on BDH without random oracles. In both constructions, the length of ciphertexts and private keys, as well as the time needed for decryption and encryption, grows linearly in the depth ℓ of the hierarchy.

---

* Extended Abstract. Full version available on the Cryptology ePrint Archives [3].

** Supported by NSF.

There are currently two principal applications for HIBE. The first, due to Canetti, Halevi, and Katz [9], is forward secure encryption. Forward secure encryption enables users to periodically update their private keys so that a message encrypted at period n cannot be read using a private key from period $n' > n$. To provide for $T = 2^t$ time periods, the CHK construction uses a HIBE of depth t where identities are *binary* vectors of length at most t. At time n, the encryptor encrypts using the identity corresponding to the n-th node of this depth t binary tree. Consequently, using previous HIBE systems [14, 1], ciphertexts in this forward secure construction are of size $O(t)$; private keys are of size $O(t^2)$ but can be reduced to size $O(t)$ by using updateable public storage. The second application for HIBE, due to Dodis and Fazio [11], is using HIBE to convert the NNL broadcast encryption system [22] into a *public-key* broadcast system. Unfortunately, the resulting public-key broadcast system is no better than simpler constructions because ciphertext length in previous HIBE constructions is linear in the depth of the hierarchy.

Our Contribution. We present a HIBE system where the ciphertext size as well as the decryption cost are *independent* of the hierarchy depth ℓ. Ciphertexts in our HIBE system are always just three group elements and decryption requires only two bilinear map computations. Private keys in our basic system contain ℓ group elements as in previous HIBE constructions.

Our system gives a forward secure encryption system with short ciphertexts consisting of only three group elements, for any number $T = 2^t$ of time periods. With our basic HIBE system, the private key size in this forward secure encryption system is $O(t^2)$. In Section 4 we describe a hybrid system that borrows some features from the Boneh-Boyen HIBE [1] and results in a forward secure encryption scheme where private key size is reduced to $O(t^{3/2})$ and ciphertext size is $O(\sqrt{t})$. By using updateable public storage as in CHK [9], private key size in these systems can be further reduced to size $O(t)$ and $O(\sqrt{t})$ respectively. In addition, instantiating the Dodis-Fazio [11] system with our HIBE system results in a *public-key* broadcast system that is as efficient as the NNL subset difference method.

It is worth noting that private keys in our system *shrink* as the identity depth increases; this shrinkage is the opposite behavior from previous HIBE systems where private keys become larger as we descend deeper down the hierarchy tree. This behavior leads to "limited delegation" where an identity at depth k can be given a restricted private key that only lets it issue private keys to descendants of limited depth (as opposed to any descendant).

Security of our system is based on a natural assumption that is closely related to the Diffie-Hellman Inversion assumption [1, 19]. We describe the assumption in Section 2.3. In the full paper [3], we prove a lower bound on the computational complexity of the problem in the generic group model and also discuss its relation to existing assumptions in bilinear groups. We present the system in Section 3 and prove its security in the selective identity model without using random oracles. We then observe that a selective-ID secure HIBE results in a fully secure HIBE in the random oracle model. In Sections 4 and 5 we discuss several extensions and applications of the system. For example, in addition to the applications already mentioned, we show how private keys can be further compressed to sublinear size and also describe an efficient mechanism for encrypting to the future.

2 Preliminaries

We briefly review the definition of HIBE and bilinear groups, and introduce the Bilinear Diffie-Hellman Exponent assumption in such groups.

2.1 Fully Secure HIBE Systems

Like an Identity Based Encryption (IBE) system, a Hierarchical Identity Based Encryption (HIBE) system consists of four algorithms [17, 14, 1]: `Setup`, `KeyGen`, `Encrypt`, `Decrypt`. In HIBE, however, identities are vectors; a vector of dimension k represents an identity at depth k. The `Setup` algorithm generates system parameters, denoted by *params*, and a master key *master-key*. We refer to the *master-key* as the private key at depth 0 and note that an IBE system is a HIBE where all identities are at depth 1. Algorithm `KeyGen` takes as input an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k)$ at depth k and the private key $d_{\mathsf{ID}|k-1}$ of the parent identity $\mathsf{ID}_{|k-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{k-1})$ at depth $k-1$, and then outputs the private key d_{ID} for identity ID. The encryption algorithm encrypts messages for an identity using *params* and the decryption algorithm decrypts ciphertexts using the private key.

Chosen ciphertext security for HIBE systems is defined under a chosen identity attack where the adversary is allowed to adaptively chose the public key on which it will be challenged. More precisely, HIBE security (IND-ID-CCA) is defined by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

Setup: The challenger $\mathcal{C}$ runs the `Setup` algorithm and gives $\mathcal{A}$ the resulting system parameters *params*, keeping the *master-key* to itself.

Phase 1: $\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_m$ where query q_i is one of the following:

- Private key query $\langle \mathsf{ID}_i \rangle$. $\mathcal{C}$ responds by running algorithm `KeyGen` to generate the private key d_i corresponding to the public key $\langle \mathsf{ID}_i \rangle$ and sends d_i to $\mathcal{A}$.
- Decryption query $\langle \mathsf{ID}_i, C_i \rangle$. $\mathcal{C}$ responds by running algorithm `KeyGen` to generate the private key d corresponding to ID_i. It then runs algorithm `Decrypt` to decrypt the ciphertext C_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 an identity ID^* and two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ on which it wishes to be challenged. The only restriction is that $\mathcal{A}$ did not previously issue a private key query for ID^* or a prefix of ID^*. $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and sets the challenge ciphertext to $\mathsf{CT} = \mathtt{Encrypt}(params, \mathsf{ID}^*, M_b)$, which is sent to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues additional queries $q_{m+1}, \ldots, q_n$ where q_i is one of:

– Private key query $\langle \mathsf{ID}_i \rangle$ where $\mathsf{ID}_i \neq \mathsf{ID}^*$ and ID_i is not a prefix of ID^*.
– Decryption query $\langle C_i \rangle \neq \langle C \rangle$ for ID^* or any prefix of ID^*.

In both cases, $\mathcal{C}$ responds as in Phase 1. These queries may be adaptive.

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

We refer to such an adversary $\mathcal{A}$ as an IND-ID-CCA adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking the scheme $\mathcal{E}$ as

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

The probability is over the random bits used by the challenger and the adversary.

Canetti, Halevi, and Katz [9, 10] define a weaker notion of security in which the adversary commits ahead of time to the public key it will attack. We refer to this notion as selective identity, chosen ciphertext secure HIBE (IND-sID-CCA). The game is exactly the same as IND-ID-CCA except that the adversary $\mathcal{A}$ discloses to the challenger the target identity ID^* before the **Setup** phase. The restrictions on private key queries from phase 2 also hold in phase 1.

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

Semantic Security. As usual, we define chosen plaintext security for a HIBE system as in the preceding game, except that the adversary is not allowed to issue any decryption queries. The adversary may still issue adaptive private key queries. This security notion is termed as IND-ID-CPA (or IND-sID-CPA in the case of a selective identity adversary).

Definition 2. *We say that a HIBE system $\mathcal{E}$ is $(t, q_{\scriptscriptstyle ID}, \epsilon)$-*IND-ID-CPA *secure (resp.* IND-sID-CPA*) if $\mathcal{E}$ is $(t, q_{\scriptscriptstyle ID}, 0, \epsilon)$-*IND-ID-CCA *secure (resp.* IND-sID-CCA*).*

2.2 Bilinear Groups

We briefly review bilinear maps and bilinear map groups. We use the following notation [18, 8]:

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

Let $\mathbb{G}$ and $\mathbb{G}_1$ be two groups as above. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ with the properties:

1. Bilinearity: 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-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists both a group $\mathbb{G}_1$ and an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ as above.

2.3 Bilinear Diffie-Hellman Exponent (BDHE) Assumption

The ℓ-BDHE problem in $\mathbb{G}$ is as follows: given g, h, and $g^{(\alpha^i)}$ in $\mathbb{G}$ for $i = 1, 2, \ldots, \ell-1, \ell+1, \ldots, 2\ell$ as input, output $e(g,h)^{(\alpha^\ell)} \in \mathbb{G}_1$. Since $g^{(\alpha^\ell)}$ is missing from the list of powers, the bilinear map seems to be of no help in computing $e(g,h)^{(\alpha^\ell)}$. As a shorthand, let $y_i = g^{(\alpha^i)} \in \mathbb{G}$. An algorithm $\mathcal{A}$ has advantage ϵ in solving ℓ-BDHE in $\mathbb{G}$ if

$$\Pr\left[\mathcal{A}\big(g, h, y_1, \ldots, y_{\ell-1}, y_{\ell+1}, \ldots, y_{2\ell}\big) = e(g,h)^{(\alpha^\ell)}\right] \geq \epsilon,$$

where the probability is over the random choice of generators g, h in $\mathbb{G}$, the random choice of α in $\mathbb{Z}_p$, and the random bits used by $\mathcal{A}$. The decisional version of the ℓ-BDHE problem in $\mathbb{G}$ is defined in the usual manner. Let $\overrightarrow{y}_{g,\alpha,\ell} = (y_1, \ldots, y_{\ell-1}, y_{\ell+1}, \ldots, y_{2\ell})$. An algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving decision ℓ-BDHE in $\mathbb{G}$ if

$$\left| \Pr\left[\mathcal{B}\big(g, h, \overrightarrow{y}_{g,\alpha,\ell}, e(g,h)^{(\alpha^\ell)}\big) = 0\right] - \Pr\left[\mathcal{B}\big(g, h, \overrightarrow{y}_{g,\alpha,\ell}, T\big) = 0\right] \right| \geq \epsilon,$$

where the probability is over the random choice of generators g, h in $\mathbb{G}$, the random choice of α in $\mathbb{Z}_p$, the random choice of $T \in \mathbb{G}_1$, and the random bits consumed by $\mathcal{B}$. We refer to the distribution on the left as $\mathcal{P}_{BDHE}$ and the distribution on the right as $\mathcal{R}_{BDHE}$.

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

For conciseness we occasionally drop the t and ϵ and simply refer to the (decision) ℓ-BDHE in $\mathbb{G}$. In the full version of this paper [3], we show that a broad class of assumptions, including the ℓ-BDHE assumption, hold in generic bilinear groups [25]; we also discuss the relation between these assumptions. We show that the ℓ-BDHE is a natural extension of the Bilinear Diffie-Hellman Inversion problem, which was previously used in various constructions [1, 12, 19].

3 A HIBE System with Constant Size Ciphertext

Let $\mathbb{G}$ be a bilinear group of prime order p and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be a bilinear map. For now, we assume that public keys (that is, identities ID) at depth k are vectors of elements in $(\mathbb{Z}_p^*)^k$. We write $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k) \in (\mathbb{Z}_p^*)^k$. The j-th component corresponds to the identity at level j. We later extend the construction to public keys over $\{0,1\}^*$ by first hashing each component I_j using a collision resistant hash $H : \{0,1\}^* \to \mathbb{Z}_p^*$. We also assume that the messages to be encrypted are elements in $\mathbb{G}_1$. The HIBE system works as follows:

`Setup`(ℓ): To generate system parameters for an HIBE of maximum depth ℓ, select a random generator $g \in \mathbb{G}$, a random $\alpha \in \mathbb{Z}_p$, and set $g_1 = g^\alpha$. Next, pick random elements $g_2, g_3, h_1, \ldots, h_\ell \in \mathbb{G}$. The public parameters and the master key are

$$\textit{params} = (g, g_1, g_2, g_3, h_1, \ldots, h_\ell), \qquad \textit{master-key} = g_2^\alpha.$$

$\texttt{KeyGen}(d_{\mathsf{ID}|k-1}, \mathbf{ID})$**:** To generate a private key d_{ID} for identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k) \in (\mathbb{Z}_p^*)^k$ of depth $k \le \ell$, pick a random $r \in \mathbb{Z}_p$ and output

$$d_{\mathsf{ID}} = \Big(g_2^\alpha \cdot \big(h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} \cdot g_3\big)^r, \quad g^r, \quad h_{k+1}^r, \quad \ldots, \quad h_\ell^r \Big) \in \mathbb{G}^{2+\ell-k}.$$

Note that d_{ID} becomes shorter as the depth of ID increases. The private key for ID can be generated just given a private key for $\mathsf{ID}_{|k-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{k-1}) \in (\mathbb{Z}_p^*)^{k-1}$ as required. Indeed, let

$$d_{\mathsf{ID}|k-1} = \Big(g_2^\alpha \cdot \big(h_1^{\mathrm{I}_1} \cdots h_{k-1}^{\mathrm{I}_{k-1}} \cdot g_3\big)^{r'}, \quad g^{r'}, \quad h_k^{r'}, \ldots, h_\ell^{r'} \Big) = (a_0, a_1, b_k, \ldots, b_\ell)$$

be the private key for $\mathsf{ID}_{|k-1}$. To generate d_{ID}, pick a random $t \in \mathbb{Z}_p$ and output

$$d_{\mathsf{ID}} = \Big(a_0 \cdot b_k^{\mathrm{I}_k} \cdot \big(h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} \cdot g_3\big)^t, \quad a_1 \cdot g^t, \quad b_{k+1} \cdot h_{k+1}^t, \quad \ldots, \quad b_\ell \cdot h_\ell^t \Big).$$

This private key is a properly distributed private key for $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k)$ for $r = r' + t \in \mathbb{Z}_p$.

$\texttt{Encrypt}(\boldsymbol{params}, \mathbf{ID}, \mathbf{M})$**:** To encrypt a message $M \in \mathbb{G}_1$ under the public key $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k) \in (\mathbb{Z}_p^*)^k$, pick a random $s \in \mathbb{Z}_p$ and output

$$\mathsf{CT} = \Big(e(g_1, g_2)^s \cdot M, \quad g^s, \quad \big(h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} \cdot g_3\big)^s \Big) \in \mathbb{G}_1 \times \mathbb{G}^2.$$

$\texttt{Decrypt}(d_{\mathsf{ID}}, \mathbf{CT})$**:** Consider an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k)$. To decrypt a given ciphertext $\mathsf{CT} = (A, B, C)$ using the private key $d_{\mathsf{ID}} = (a_0, a_1, b_{k+1} \ldots, b_\ell)$, output

$$A \cdot e(a_1, C) \Big/ e(B, a_0) = M.$$

Indeed, for a valid ciphertext, we have

$$\frac{e(a_1, C)}{e(B, a_0)} = \frac{e\Big(g^r, \ (h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} \cdot g_3)^s\Big)}{e\Big(g^s, \ g_2^\alpha (h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} \cdot g_3)^r\Big)} = \frac{1}{e(g, g_2)^{s\alpha}} = \frac{1}{e(g_1, g_2)^s}.$$

Observe that for identities at any depth, the ciphertext contains only 3 elements and decryption takes only 2 pairings. In previous HIBE systems, ciphertext size and decryption time grow linearly in the identity depth. Also, note that $e(g_1, g_2)$ used for encryption can be precomputed (or substituted for g_2 in the system parameters) so that encryption does not require any pairings.

3.1 Security

We first show that our HIBE scheme is selective identity secure (IND-sID-CPA) under the decisional Bilinear Diffie-Hellman Exponent assumption. We later describe how to provide both chosen ciphertext security (IND-sID-CCA) and full HIBE security (IND-ID-CCA).

Theorem 1. *Let $\mathbb{G}$ be a bilinear group of prime order p. Suppose the decision $(t, \epsilon, \ell+1)$-BDHE assumption holds in $\mathbb{G}$. Then the previously defined ℓ-HIBE system is (t', q_S, ϵ)-selective identity, chosen plaintext* (IND-sID-CPA) *secure for arbitrary ℓ, q_S, and $t' < t - \Theta(\tau\, \ell\, q_S)$, where τ is the maximum time for an exponentiation in $\mathbb{G}$.*

Proof. Suppose $\mathcal{A}$ has advantage ϵ in attacking the ℓ-HIBE system. Using $\mathcal{A}$, we build an algorithm $\mathcal{B}$ that solves the decision $(\ell+1)$-BDHE problem in $\mathbb{G}$.

For a generator $g \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$ let $y_i = g^{(\alpha^i)} \in \mathbb{G}$. Algorithm $\mathcal{B}$ is given as input a random tuple $(g, h, y_1, \ldots, y_\ell, y_{\ell+2}, \ldots, y_{2\ell+2}, T)$ that is either sampled from $\mathcal{P}_{BDHE}$ (where $T = e(g,h)^{(\alpha^{\ell+1})}$) or from $\mathcal{R}_{BDHE}$ (where T is uniform and independent in $\mathbb{G}_1$). Algorithm $\mathcal{B}$'s goal is to output 1 when the input tuple is sampled from $\mathcal{P}_{BDHE}$ and 0 otherwise. Algorithm $\mathcal{B}$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

Initialization. The selective identity game begins with $\mathcal{A}$ first outputting an identity $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_m^*) \in (\mathbb{Z}_p^*)^m$ of depth $m \le \ell$ that it intends to attack. If $m < \ell$ then $\mathcal{B}$ pads ID^* with $\ell - m$ zeroes on the right to make ID^* a vector of length ℓ. Hence, from here we assume that ID^* is a vector of length ℓ.

Setup. To generate the system parameters, algorithm $\mathcal{B}$ picks a random γ in $\mathbb{Z}_p$ and sets $g_1 = y_1 = g^\alpha$ and $g_2 = y_\ell \cdot g^\gamma = g^{\gamma + (\alpha^\ell)}$. Next, $\mathcal{B}$ picks random $\gamma_1, \ldots, \gamma_\ell$ in $\mathbb{Z}_p$ and sets $h_i = g^{\gamma_i}/y_{\ell-i+1}$ for $i = 1, \ldots, \ell$. Algorithm $\mathcal{B}$ also picks a random δ in $\mathbb{Z}_p$ and sets $g_3 = g^\delta \cdot \prod_{i=1}^{\ell} y_{\ell-i+1}^{I_i^*}$.

Finally, $\mathcal{B}$ gives $\mathcal{A}$ the system parameters $params = (g, g_1, g_2, g_3, h_1, \ldots, h_\ell)$. Observe that all these values are distributed uniformly and independently in $\mathbb{G}$ as required. The master key corresponding to these system parameters is $g_2^\alpha = g^{\alpha(\alpha^\ell + \gamma)} = y_{\ell+1} y_1^\gamma$, which is unknown to $\mathcal{B}$ since $\mathcal{B}$ does not have $y_{\ell+1}$.

Phase 1. $\mathcal{A}$ issues up to q_S private key queries. Consider a query for the private key corresponding to $\mathsf{ID} = (\mathrm{I}_1, \ldots, I_u) \in (\mathbb{Z}_p^*)^u$ where $u \le \ell$. The only restriction is that ID is not ID^* or a prefix of ID^*. This restriction ensures that there exists a $k \in \{1, \ldots, u\}$ such that $\mathrm{I}_k \ne \mathrm{I}_k^*$ (otherwise, ID would be a prefix of ID^*). To respond to the query, algorithm $\mathcal{B}$ first derives a private key for the identity $(\mathrm{I}_1, \ldots, \mathrm{I}_k)$ from which it then constructs a private key for the requested identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k, \ldots, \mathrm{I}_u)$.

To generate the private key for identity $(\mathrm{I}_1, \ldots, \mathrm{I}_k)$, $\mathcal{B}$ first picks a random $\tilde{r}$ in $\mathbb{Z}_p$. We pose $r = \frac{\alpha^k}{(\mathrm{I}_k - \mathrm{I}_k^*)} + \tilde{r} \in \mathbb{Z}_p$. Next, $\mathcal{B}$ generates the private key

$$\left(g_2^\alpha \cdot (h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} g_3)^r, \;\; g^r, \;\; h_{k+1}^r, \;\; \ldots, \;\; h_\ell^r \right), \tag{1}$$

which is a properly distributed private key for the identity $(\mathrm{I}_1, \ldots, \mathrm{I}_k)$. We show that $\mathcal{B}$ can compute all elements of this private key given the values at its disposal. We use the fact that $y_i^{(\alpha^j)} = y_{i+j}$ for any i, j.

To generate the first component of the private key, first observe that

$$(h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} g_3)^r = \left(g^{\delta + \sum_{i=1}^{k} \mathrm{I}_i \gamma_i} \cdot \prod_{i=1}^{k-1} y_{\ell-i+1}^{(\mathrm{I}_i^* - \mathrm{I}_i)} \cdot y_{\ell-k+1}^{(\mathrm{I}_k^* - \mathrm{I}_k)} \cdot \prod_{i=k+1}^{\ell} y_{\ell-i+1}^{\mathrm{I}_i^*} \right)^r. \qquad (2)$$

Let Z denote the product of the first, second, and fourth terms. That is,

$$Z = \left(g^{\delta + \sum_{i=1}^{k} \mathrm{I}_i \gamma_i} \cdot \prod_{i=1}^{k-1} y_{\ell-i+1}^{(\mathrm{I}_i^* - \mathrm{I}_i)} \cdot \prod_{i=k+1}^{\ell} y_{\ell-i+1}^{\mathrm{I}_i^*} \right)^r.$$

One can verify that $\mathcal{B}$ can compute all the terms in Z given the values at its disposal. Next, observe that the third term in Eq (2), namely $y_{\ell-k+1}^{r(\mathrm{I}_k^* - \mathrm{I}_k)}$, is:

$$y_{\ell-k+1}^{r(\mathrm{I}_k^* - \mathrm{I}_k)} = y_{\ell-k+1}^{\tilde{r}(\mathrm{I}_k^* - \mathrm{I}_k)} \cdot y_{\ell-k+1}^{(\mathrm{I}_k^* - \mathrm{I}_k)\frac{\alpha^k}{(\mathrm{I}_k - \mathrm{I}_k^*)}} = y_{\ell-k+1}^{\tilde{r}(\mathrm{I}_k^* - \mathrm{I}_k)} / y_{\ell+1}.$$

Hence, the first component in the private key (1) is equal to:

$$g_2^{\alpha}(h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} g_3)^r = (y_{\ell+1} y_1^{\gamma}) \cdot Z \cdot (y_{\ell-k+1}^{\tilde{r}(\mathrm{I}_k^* - \mathrm{I}_k)} / y_{\ell+1}) = y_1^{\gamma} \cdot Z \cdot y_{\ell-k+1}^{\tilde{r}(\mathrm{I}_k^* - \mathrm{I}_k)}.$$

Since $y_{\ell+1}$ cancels out, all the terms in this expression are known to $\mathcal{B}$. Thus, $\mathcal{B}$ can compute the first private key component.

The second component, g^r, is $y_k^{1/(\mathrm{I}_k - \mathrm{I}_k^*)} g^{\tilde{r}}$ which $\mathcal{B}$ can compute. Similarly, the remaining elements $h_{k+1}^r, \ldots, h_\ell^r$ can be computed by $\mathcal{B}$ since they do not involve a $y_{\ell+1}$ term. Thus, $\mathcal{B}$ can derive a valid private key for $(\mathrm{I}_1, \ldots, \mathrm{I}_k)$. Algorithm $\mathcal{B}$ uses this private key to derive a private key for the descendant identity ID and gives $\mathcal{A}$ the result.

Challenge. When $\mathcal{A}$ decides that Phase 1 is over, it outputs two messages $M_0, M_1 \in \mathbb{G}_1$ on which it wishes to be challenged. Algorithm $\mathcal{B}$ picks a random bit $b \in \{0, 1\}$ and responds with the challenge ciphertext

$$\mathsf{CT} = (M_b \cdot T \cdot e(y_1, h^{\gamma}), \;\; h, \;\; h^{\delta + \sum_{i=1}^{\ell} \mathrm{I}_i^* \gamma_i})$$

where h and T are from the input tuple given to $\mathcal{B}$. First note that if $h = g^c$ (for some unknown c in $\mathbb{Z}_p$) then

$$h^{\delta + \sum_{i=1}^{\ell} \mathrm{I}_i^* \gamma_i} = \left(\prod_{i=1}^{\ell} (g^{\gamma_i} / y_{\ell-i+1})^{\mathrm{I}_i^*} \cdot (g^{\delta} \prod_{i=1}^{\ell} y_{\ell-i+1}^{\mathrm{I}_i^*}) \right)^c = (h_1^{\mathrm{I}_1^*} \cdots h_\ell^{\mathrm{I}_\ell^*} g_3)^c, \quad \text{and}$$

$$e(g, h)^{(\alpha^{\ell+1})} \cdot e(y_1, h^{\gamma}) = \big(e(y_1, y_\ell) \cdot e(y_1, g^{\gamma}) \big)^c = e(y_1, y_\ell g^{\gamma})^c = e(g_1, g_2)^c.$$

Therefore, if $T = e(g, h)^{(\alpha^{\ell+1})}$ (i.e., when the input tuple is sampled from $\mathcal{P}_{BDHE}$), then the challenge ciphertext is a valid encryption of M_b under the original (unpadded) identity $\mathsf{ID}^* = (\mathrm{I}_1^*, \ldots, \mathrm{I}_m^*)$ chosen by the adversary, since

$$\begin{aligned}\mathsf{CT} &= \big(\ M_b \cdot e(g_1, g_2)^c, \quad g^c, \quad (h_1^{\mathrm{I}_1^*} \cdots h_m^{\mathrm{I}_m^*} \cdots h_\ell^{\mathrm{I}_\ell^*} g_3)^c\ \big)\\ &= \big(\ M_b \cdot e(g_1, g_2)^c, \quad g^c, \quad (h_1^{\mathrm{I}_1^*} \cdots h_m^{\mathrm{I}_m^*} g_3)^c\ \big).\end{aligned}$$

On the other hand, when T is uniform and independent in $\mathbb{G}_1$ (when the input tuple is sampled from $\mathcal{R}_{BDHE}$), CT is independent of b in the adversary's view.

Phase 2. $\mathcal{A}$ issues queries not issued in Phase 1. $\mathcal{B}$ responds as before.

Guess. Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. Algorithm $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $b = b'$ then $\mathcal{B}$ outputs 1 meaning $T = e(g, h)^{(\alpha^{\ell+1})}$. Otherwise, it outputs 0 meaning T is random in $\mathbb{G}_1$.

When the input tuple is sampled from $\mathcal{P}_{BDHE}$ (where $T = e(g, h)^{(\alpha^{\ell+1})}$), then $\mathcal{A}$'s view is identical to its view in a real attack game and therefore $\mathcal{A}$ satisfies $|\Pr[b = b'] - 1/2| \geq \epsilon$. When the input tuple is sampled from $\mathcal{R}_{BDHE}$ (where T is uniform in $\mathbb{G}_1$) then $\Pr[b = b'] = 1/2$. Therefore, with g, h uniform in $\mathbb{G}$, α uniform in $\mathbb{Z}_p$, and T uniform in $\mathbb{G}_1$ we have that

$$\begin{aligned}\Big| \Pr\Big[\mathcal{B}\big(g, h, \overrightarrow{y}_{g,\alpha,\ell}, e(g, h)^{(\alpha^{\ell+1})}\big) = 0\Big] \quad - \quad \Pr\Big[\mathcal{B}\big(g, h, \overrightarrow{y}_{g,\alpha,\ell}, T\big) = 0\Big]\Big| \\ \geq \quad |(1/2 \pm \epsilon) - 1/2| \ = \ \epsilon\end{aligned}$$

as required. This completes the proof of the theorem.

Chosen Ciphertext Security. Canetti et al. [10] show a general method of building an $\mathsf{IND\text{-}sID\text{-}CCA}$ secure ℓ-HIBE from a $\mathsf{IND\text{-}sID\text{-}CPA}$ secure $\ell + 1$-HIBE. A more efficient construction is given by Boneh and Katz [7]. Applying either method to our HIBE construction results in a $\mathsf{IND\text{-}sID\text{-}CCA}$ secure ℓ-HIBE for arbitrary ℓ where the ciphertext length is independent of the hierarchy height.

Arbitrary Identities. We can extend our HIBE to handle arbitrary identities $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_\ell)$ with $\mathrm{I}_i \in \{0, 1\}^*$ for $i = 1, \ldots, \ell$ by hashing each I_i with a collision resistant hash function $H : \{0, 1\}^* \to \mathbb{Z}_p^*$ during key generation and encryption. A standard argument shows that if the original HIBE scheme is $\mathsf{IND\text{-}sID\text{-}CCA}$ secure, then so is the HIBE scheme using H.

3.2 Full HIBE Security

Theorem 1 shows that our HIBE system is selective-ID secure without random oracles. Thus, the system is secure when the adversary commits ahead of time to the identity he intends to attack. Boneh and Boyen [1] observed that IBE systems that are selective-ID secure are also fully secure (i.e., secure against adversaries that adaptively select the identity to attack) as long as one hashes the identity prior to using it. The reduction, however, is not tight. Let $H : \{0, 1\}^* \to \{0, 1\}^d$ be a hash function (where, e.g., $d = 160$ bits). Assuming H is collision resistant, the reduction introduces a 2^d multiplicative security loss factor in the standard model. When H is viewed as a random oracle, the reduction introduces a q_H multiplicative security loss factor where q_H is the number of the hash oracle queries.

A similar observation applies to HIBE systems. Let $\mathcal{E}$ be a selective-ID secure HIBE of depth ℓ. Let $\mathcal{E}_H$ be an HIBE system where an identity $\mathsf{ID} = (I_1, \ldots, I_k)$ is hashed to $\mathsf{ID}_H = (H(I_1), \ldots, H(I_k))$ before using it in `KeyGen` and `Encrypt`. Then, if H is collision resistant, it follows that $\mathcal{E}_H$ is a fully secure HIBE, but the reduction introduces a loss factor of $2^{\ell d}$. In the random oracle model, $\mathcal{E}_H$ is a fully secure HIBE and the reduction introduces a loss factor of q_H^ℓ.

We remark that in the random oracle model, the public parameters are of constant size and contain only the two group elements (g, g_1); the other parameters $(g_2, g_3, h_1, \ldots, h_\ell)$ need not be specified as they can be derived by applying the oracle on a predetermined input string.

We also note that the construction of Waters [26], for a fixed depth ℓ, applied to our HIBE could give a constant ciphertext HIBE with a polynomial time reduction to the underlying complexity assumption. The resulting private keys are much larger, namely of size $d\ell$, as opposed to ℓ in our system.

4 Extensions

We discuss a number of extensions to the HIBE system of the previous section.

4.1 Limited Delegation

Let $d_{\mathsf{ID}} = (a_0, a_1, b_k, \ldots, b_\ell)$ be the private key for the identity ID. Note that the `Decrypt` algorithm uses only the terms a_0 and a_1, and the `KeyGen` algorithm uses only the remaining terms $b_k, \ldots, b_\ell$.

By removing any number of $b_k, \ldots, b_\ell$, an identity ID at depth k can be given a restricted private key that only lets it issue private keys to descendants of bounded depth. For example, if the private key for ID only contains b_k, b_{k+1}, b_{k+2} (instead of all $b_k, \ldots, b_\ell$), then ID can only issue private keys for three generations of descendants, and those descendants' private keys will be limited even further.

4.2 HIBE with Short Private Keys

Certain applications, such as the time lock encryption (to be described in Section 5), are better served by using a HIBE system with short private keys rather than ciphertexts. We show how to construct a HIBE system whose private key size grows only sublinearly with hierarchy depth.

The idea is to construct a hybrid of the HIBE in Section 3 and the Boneh-Boyen HIBE [1]. Recall that in the former system the private key shrinks as the identity depth increases, while in the latter system the private key grows with the depth of an identity. The hybrid is based on the algebraic similarities between both systems, and exploits their opposite behavior with regard to private key size, to ensure that no private key ever contains more than $O(\sqrt{\ell})$ group elements.

Specifically, for $\omega \in [0,1]$, the hybrid scheme achieves $O(\ell^{\omega} + \ell^{1-\omega})$ private key size and $O(\ell^{\omega})$ ciphertext size at every level in a hierarchy of depth ℓ. The setting $\omega = 0$ corresponds to our HIBE, whereas $\omega = 1$ corresponds to the Boneh-Boyen HIBE [1]. The most efficient hybrids are obtained when $\omega \in [0, 1/2]$. For example, when $\omega = 1/2$, private keys and ciphertexts are of size $O(\sqrt{\ell})$.

Hybrid Scheme. As before, we assume a bilinear group $\mathbb{G}$ and a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$, where $\mathbb{G}$ and $\mathbb{G}_1$ have prime order p. Let $\ell_1 = \lceil \ell^{\omega} \rceil$ and $\ell_2 = \lceil \ell^{1-\omega} \rceil$. The basic idea is to partition levels of the hierarchy into ℓ_1 consecutive groups of size ℓ_2. Within each group we use the system of Section 3. Between groups we use the Boneh-Boyen HIBE [1].

Let $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k) \in (\mathbb{Z}_p^*)^k$ be an identity of depth $k \le \ell$. We will represent ID as a pair (k, I) where $\mathrm{I} \in (\mathbb{Z}_p^*)^{\ell_1 \times \ell_2}$ is an $\ell_1 \times \ell_2$ matrix filled using the elements $\mathrm{I}_1, \ldots, \mathrm{I}_k$ in typographic order: one row at a time starting from the top, in each row starting from the left (note that $\ell_1 \cdot \ell_2 \ge \ell \ge k$; the unfilled matrix entries are undefined). For convenience, we decompose the indices $k = 1, \ldots, \ell$ into row-column pairs (k_1, k_2) such that $k = \ell_2 \cdot (k_1 - 1) + k_2$ where $k_1, k_2 > 0$. For shorthand, we write $(k_1, k_2) = k$. It follows that in the above matrix representation of ID we have $\mathrm{I}_{(i_1,i_2)} = \mathrm{I}_i$ for all $i = 1, \ldots, k$. Or, pictorially, for an ID at the maximum depth ℓ with $\mathrm{I} = \mathrm{I}_1, \ldots, \mathrm{I}_\ell$ and $\ell = \ell_1 \ell_2$:

$$\mathrm{I} \;=\; \begin{pmatrix} \mathrm{I}_1 & \mathrm{I}_2 & \ldots & \mathrm{I}_{\ell_2} \\ \mathrm{I}_{\ell_2+1} & \mathrm{I}_{\ell_2+2} & \ldots & \mathrm{I}_{2\,\ell_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{I}_{(\ell_1-1)\ell_2+1} & \mathrm{I}_{(\ell_1-1)\ell_2+2} & \ldots & \mathrm{I}_{\ell_1\ell_2} \end{pmatrix} \;=\; \begin{pmatrix} \mathrm{I}_{(1,1)} & \mathrm{I}_{(1,2)} & \ldots & \mathrm{I}_{(1,\ell_2)} \\ \mathrm{I}_{(2,1)} & \mathrm{I}_{(2,2)} & \ldots & \mathrm{I}_{(2,\ell_2)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{I}_{(\ell_1,1)} & \mathrm{I}_{(\ell_1,2)} & \ldots & \mathrm{I}_{(\ell_1,\ell_2)} \end{pmatrix}.$$

Using this convention, we can now describe the hybrid HIBE system as follows.

$\mathtt{Setup}(\ell, \omega)$**:** For a HIBE of maximum depth ℓ, first determine ℓ_1 and ℓ_2 as above so that $\ell \le \ell_1 \cdot \ell_2$. Next, select a random generator g in $\mathbb{G}$, a random $\alpha \in \mathbb{Z}_p$, and set $g_1 = g^{\alpha}$. Then, pick random elements $g_2, f_1, \ldots, f_{\ell_1}, h_1, \ldots, h_{\ell_2} \in \mathbb{G}$. The public parameters *params* and the secret *master-key* are given by

$$\mathit{params} = (\; g,\; g_1, g_2,\; f_1, \ldots, f_{\ell_1},\; h_1, \ldots, h_{\ell_2}\;)\;, \qquad \mathit{master\text{-}key} = g_2^{\alpha}.$$

$\mathtt{KeyGen}(d_{\mathsf{ID}|k-1}, \mathsf{ID})$**:** To generate private key d_{ID} for identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k) \in (\mathbb{Z}_p^*)^k$ of depth $(k_1, k_2) = k \le \ell$, where $k_1 \le \ell_1$ and $k_2 \le \ell_2$, pick random $r_1, \ldots, r_{k_1} \in \mathbb{Z}_p$, and output

$$\begin{aligned} d_{\mathsf{ID}} \;=\; \Bigg(& g_2^{\alpha} \cdot \left(\prod_{i=1}^{k_1-1} \left(h_1^{\mathrm{I}_{(i,1)}} \cdots h_{\ell_2}^{\mathrm{I}_{(i,\ell_2)}} \cdot f_i \right)^{r_i} \right) \cdot \left(h_1^{\mathrm{I}_{(k_1,1)}} \cdots h_{k_2}^{\mathrm{I}_{(k_1,k_2)}} \cdot f_{k_1} \right)^{r'_{k_1}}, \\ & g^{r_1},\; \ldots,\; g^{r_{k_1-1}},\; g^{r'_{k_1}},\; h_{k_2+1}^{r'_{k_1}},\; \ldots,\; h_{\ell_2}^{r'_{k_1}} \Bigg) \;\in\; \mathbb{G}^{1+k_1+\ell_2-k_2}. \end{aligned} \tag{3}$$

Note that the factors $(\ldots)^{r_i}$ under the $\prod$ sign contain ℓ_2 identity terms each, whereas the last factor $(\ldots)^{r_{k_1}}$ only has k_2 such terms. The size of d_{ID} grows with

k_1 and shrinks with k_2; the private key thus becomes alternatively shorter and longer as the depth of ID increases, but never exceeds $\ell_1 + \ell_2$ elements of $\mathbb{G}$.

The private key for ID can be generated with a private key for $\mathsf{ID}_{|k-1} = (\mathrm{I}_1, \ldots, \mathrm{I}_{k-1}) \in (\mathbb{Z}_p^*)^{k-1}$ as required. Decompose k as (k_1, k_2) according to our convention. There are two cases:

1. If $k-1$ is written $(k_1, k_2 - 1)$, namely k and $k-1$ have the same row index k_1, then we know that the private key for $\mathsf{ID}_{|k-1}$ is of the form:

$$d_{\mathsf{ID}|k-1} = \Bigg(g_2^{\alpha} \cdot \prod_{i=1}^{k_1-1} \left(h_1^{\mathrm{I}_{(i,1)}} \cdots h_{\ell_2}^{\mathrm{I}_{(i,\ell_2)}} \cdot f_i\right)^{r_i} \cdot \left(h_1^{\mathrm{I}_{(k_1,1)}} \cdots h_{k_2-1}^{\mathrm{I}_{(k_1,k_2-1)}} \cdot f_{k_1}\right)^{r_{k_1}}, \;\; g^{r_1},$$
$$\ldots, \;\; g^{r_{k_1}}, h_{k_2}^{r_{k_1}}, \;\; \ldots, \;\; h_{\ell_2}^{r_{k_1}} \Bigg) \; = \; (a_0, b_1, \ldots, b_{k_1}, c_{k_2}, \ldots, c_{\ell_2}) \;\; \in \mathbb{G}^{2+k_1+\ell_2-k_2}.$$

In this case, to generate d_{ID} from $d_{\mathsf{ID}|k-1}$, pick a random $r^* \in \mathbb{Z}_p$ and output

$$d_{\mathsf{ID}} = \Bigg(a_0 \cdot c_{k_2}^{\mathrm{I}_{(k_1,k_2)}} \cdot \left(h_1^{\mathrm{I}_{(k_1,1)}} \cdots h_{k_2}^{\mathrm{I}_{(k_1,k_2)}} \cdot f_{k_1}\right)^{r^*}, \;\; b_1, \;\; \ldots, \;\; b_{k_1-1}, \;\; b_{k_1} \cdot g^{r^*},$$
$$c_{k_2+1} \cdot h_{k_2+1}^{r^*}, \;\; \ldots, \;\; c_{\ell_2} \cdot h_{\ell_2}^{r^*} \Bigg) \;\; \in \mathbb{G}^{1+k_1+\ell_2-k_2}.$$

This tuple is of the same form as Eq (3) where $r'_{k_1} = r_{k_1} + r^*$.

2. If the row indices differ, then necessarily $k-1 = (k_1 - 1, \ell_2)$ and $k = (k_1, 1)$, and the private key for $\mathsf{ID}_{|k-1}$ must be of the form:

$$d_{\mathsf{ID}|k-1} = \Bigg(g_2^{\alpha} \cdot \prod_{i=1}^{k_1-1} \left(h_1^{\mathrm{I}_{(i,1)}} \cdots h_{\ell_2}^{\mathrm{I}_{(i,\ell_2)}} \cdot f_i\right)^{r_i}, \;\; g^{r_1}, \;\; \ldots, \;\; g^{r_{k_1-1}} \Bigg)$$
$$= (a_0, b_1, \ldots, b_{k_1-1}) \;\; \in \mathbb{G}^{k_1}.$$

In this case, to generate d_{ID} from $d_{\mathsf{ID}|k-1}$, pick a random $r \in \mathbb{Z}_p$ and output

$$d_{\mathsf{ID}} = \Big(a_0 \cdot \left(h_1^{\mathrm{I}_{(k_1,1)}} \cdot f_{k_1}\right)^{r}, \;\; b_1, \;\; \ldots, \;\; b_{k_1-1}, \;\; g^r, \;\; h_2^r, \;\; , \ldots, h_{\ell_2}^r \Big) \;\; \in \mathbb{G}^{k_1+\ell_2}.$$

Again, this tuple conforms to Eq (3) in which r_{k_1} has been set to r.

`Encrypt(`***params*, ID, M**`)`**:** To encrypt a message $M \in \mathbb{G}_1$ under the public key $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k) \in \mathbb{Z}_p^k$ where $k = (k_1, k_2)$, pick a random $s \in \mathbb{Z}_p$ and output

$$\mathsf{CT} = \Big(\; e(g_1, g_2)^s \cdot M, \;\; g^s, \;\; \left(h_1^{\mathrm{I}_{(1,1)}} \cdots h_{\ell_2}^{\mathrm{I}_{(1,\ell_2)}} \cdot f_1\right)^s, \;\; \ldots \;\; ,$$
$$\left(h_1^{\mathrm{I}_{(k_1-1,1)}} \cdots h_{\ell_2}^{\mathrm{I}_{(k_1-1,\ell_2)}} \cdot f_{k_1-1}\right)^s, \;\; \left(h_1^{\mathrm{I}_{(k_1,1)}} \cdots h_{k_2}^{\mathrm{I}_{(k_1,k_2)}} \cdot f_{k_1}\right)^s \; \Big) \;\; \in \mathbb{G}_1 \times \mathbb{G}^{1+k_1}.$$

`Decrypt(`d_{ID}, **CT**`)`**:** Consider an identity $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_k)$ with $k = (k_1, k_2)$. To decrypt a ciphertext $\mathsf{CT} = (A, B, C_1, \ldots, C_{k_1-1}, C_{k_1})$ using the private key $d_{\mathsf{ID}} =$

$(a_0, b_1, \ldots, b_{k_1}, c_{k_2+1}, \ldots, c_{\ell_2})$, output

$$A \cdot \prod_{i=1}^{k_1} e(b_i, C_i) \Big/ e(B, a_0) = M.$$

Note that the private key components $c_{k_2+1}, \ldots, c_{\ell_2}$ are not used for decryption.

Complexity. It is easy to see that in a hierarchy of depth ℓ, private keys at all levels contain at most $\ell_1 + \ell_2$ elements of $\mathbb{G}$, while ciphertexts contain at most $1 + \ell_1$ elements of $\mathbb{G}$ and one element of $\mathbb{G}_1$. Encryption, decryption, and one-level-down key generation, all require $O(\ell_1 + \ell_2)$ operations, or $O(\sqrt{\ell})$ for the choice $\omega = 1/2$ as claimed. We note that the combination of having a selectable parameter ω together with the option of using an asymmetric bilinear group geared toward reducing the ciphertext or the private key size (described in Section 4.3), gives great flexibility toward achieving the optimal trade-off for a given application.

Security. We prove security based on the $(\ell_2 + 1)$-BDHE assumption (observe that the BDHE assumption implies the BDH assumption). We note that for $\omega = 1/2$, security for a ℓ-level hierarchy is based on the $O(\sqrt{\ell})$-BDHE assumption.

Theorem 2. *Let $\mathbb{G}$ be a bilinear group of prime order p. Consider a hybrid ℓ-HIBE system with identity hierarchy partitioned into ℓ_1 groups each of size ℓ_2. Suppose the decision $(t, \epsilon, \ell_2 + 1)$-BDHE assumption holds in $\mathbb{G}$. Then the hybrid ℓ-HIBE system is (t', q_S, ϵ)-selective identity, chosen plaintext* (IND-sID-CPA) *secure for arbitrary ℓ, q_S, and $t' < t - \Theta(\tau\, \ell\, q_S)$, where τ is the maximum time for an exponentiation in $\mathbb{G}$.*

The proof is similar to that for Theorem 1 and is in the full paper [3].

4.3 Asymmetric Bilinear Groups and MNT Curves.

It is often desirable to use bilinear maps $e : \mathbb{G} \times \mathbb{G}' \to \mathbb{G}_1$ where $\mathbb{G}$ and $\mathbb{G}'$ are distinct groups. Such maps let us take advantage of certain curves called MNT curves [20]. Typically, elements of the group $\mathbb{G}$ tend to afford a particularly compact representation compared to elements of $\mathbb{G}'$. This property is used for constructing short signatures [8, 2, 4]. For our system, we can use this property to shrink either the private keys or the ciphertexts. Details are in the full paper.

5 Applications

We now discuss applications of our compact HIBE system and its extensions.

5.1 Forward Secure Encryption

The main purpose of a forward secure encryption scheme is to guarantee that all messages encrypted before the secret key is compromised remain secret.

An elegant public key encryption scheme with forward security was proposed by Canetti, Halevi, and Katz (CHK) [9]. Let $T = 2^t$ be the number of distinct time periods in the forward secure system. When implemented with previous HIBE systems [14, 1], the CHK framework results in ciphertexts of size $O(t)$ and private keys of size $O(t^2)$. Using public updateable storage, Canetti et al. reduce private key size to $O(t)$ without affecting ciphertext length — the idea is to encrypt the private key for time period i under the public key of time period $i-1$ and store the resulting ciphertext, of size $O(t^2)$, in public storage; consequently, only one HIBE private key of size $O(t)$ is kept in private storage.

Using the HIBE system of Section 3 in the CHK framework, we obtain a forward secure encryption scheme with $O(1)$ ciphertext size and decryption time — independent of the number of time periods. Private keys using our basic system are of size $O(t^2)$. Alternatively, using the hybrid HIBE of Section 4.2 in which we set $\omega = 1/2$, we obtain a forward secure encryption scheme with private key size $O(t^{3/2})$; in this case ciphertext size and decryption time become $O(\sqrt{t})$.

Following Canetti et al. [9], we can store most of the private key in updateable public storage in order to lessen the private storage requirement. Applied to our basic forward secure system, using $O(t^2)$ public storage we can reduce the private key size to $O(t)$ while keeping the ciphertext size constant. Using the hybrid HIBE system (for $\omega = 1/2$), the private storage requirement can be similarly reduced to $O(\sqrt{t})$ at the cost of $O(t^{3/2})$ updateable public storage; ciphertext size in this case remains $O(\sqrt{t})$.

5.2 Forward Secure HIBE

Recently, a forward secure HIBE scheme was proposed by Yao et al. [27]. Their scheme essentially uses two HIBE hierarchies in the manner of Canetti et al. [9] to obtain forward security together with the ability to derive subordinate keys. Their system has ciphertexts of size $O(\ell \cdot t)$ where ℓ is the depth of the identity hierarchy and $T = 2^t$ is the number of time periods. Indeed, they pose as an open problem if a forward secure HIBE scheme with "linear" complexity is possible.

Instantiating both hierarchies in their construction with our HIBE system immediately gives a forward secure HIBE scheme with ciphertexts of size $O(1)$, which resolves this question.

We also propose a more specific forward secure HIBE construction that achieves "linear" $O(\ell + t)$ size for all components, including private keys and public parameters (ciphertexts are no longer constant size in that construction). The construction is a hybrid between the HIBE given in Section 3 and the Boneh-Boyen HIBE from [1]; it is described in detail in the full paper [3].

5.3 Public Key NNL Broadcast Encryption

Broadcast encryption schemes, introduced by Fiat and Naor [13], are cryptosystems designed for the efficient broadcast of data to a dynamic group of users authorized to receive the data. Naor, Naor, and Lotspiech [22] considered broadcast encryption in the stateless receiver setting; they provided a general "subset cover" framework for such broadcast encryption schemes and gave two instances of the framework — the Complete Subtree (CS) method and the more efficient Subset Difference (SD) method. Further improvements have been proposed such as the Layered Subset Difference (LSD) [16] and the Stratified Subset Difference (SSD) [15]. In the symmetric key setting, only a "center" that possesses the secret keys can broadcast to the users. In a public key broadcast encryption system, anyone is allowed to broadcast to selected subsets of users.

Using the HIBE framework, Dodis and Fazio [11] showed how to translate the SD and LSD methods to the public key setting. For N users and r revoked users, their SD and LSD constructions based on previous HIBE systems give ciphertexts of size $O(r \cdot \log N)$, which is no better than the basic CS method. Substituting the HIBE system of Section 3 restores the full benefits of both SD and LSD, which results in ciphertexts of size $O(r)$.

5.4 Encrypting to the Future

Mont et al. [21] observed that an IBE system gives a mechanism for encrypting to the future using a trusted server. Let D be a certain date string. We view D as a public key in an IBE system. Every day, a trusted server publishes the private key corresponding to that day, which enables messages encrypted for that day to be decrypted. Methods for encrypting to the future without a trusted server were proposed by Rivest, Shamir, and Wagner [23].

One problem with the IBE timelock mechanism is that after n days have passed, the server has to publish a bulletin board with n private keys on it (one private key for each day). The amount of data on the bulletin board can be greatly reduced by using the CHK forward secure encryption scheme *in reverse*. Suppose the CHK framework is set up for a total of T time periods (using a tree of depth $\log_2 T$). To encrypt a message for day $n < T$, use the CHK public key for time period $T - n$. Similarly, on day n the trusted server publishes the CHK private key corresponding to time period $T - n$. This single private key enables anyone to derive the private keys for CHK time periods $T - n, T - n + 1, \ldots, T$. Anyone can thus decrypt messages intended for days in the range $1, \ldots, n$.

Implementing this encoding using our $O(1)$ ciphertext HIBE, the trusted server on any day only needs to publish a single private key comprising $O(\log^2 T)$ group elements. Using the hybrid HIBE system of Section 4.2, the private key posted by the server is further reduced to $O(\log^{3/2} T)$ group elements for ciphertexts of size $O(\sqrt{\log T})$. These parameters are much better than the IBE based mechanism [21], where the bulletin board contains as many as T group elements.

6 Conclusions and Open Problems

We presented a new HIBE system where the ciphertexts consist of three group elements and decryption only requires computing two bilinear maps, both of which are independent of the hierarchy depth. Encryption time is as efficient as other HIBE systems. For a hierarchy of depth ℓ we proved security based on the

$(\ell+1)$-BDHE assumption. We expect ℓ-BDHE to be very useful for constructing cryptosystems with short ciphertexts. For example, ℓ-BDHE was recently used to construct a broadcast encryption system [6] where both ciphertexts and private keys are short.

We discussed several applications of our system, including efficient forward secure encryption, an efficient public key version of the NNL broadcast encryption system, and an efficient mechanism for encrypting to the future. Our HIBE system allows for limited delegation and can be combined with the Boneh-Boyen HIBE to form a hybrid HIBE that has sublinear private key size.

We note that our selective-ID proof of security is tight. On the other hand, the proof of full security (either in the random oracle or standard model) discussed in Section 3.2 degrades exponentially in the hierarchy depth. The same is true for all existing HIBE systems. It is an open problem to construct a HIBE system where security does not degrade exponentially in the hierarchy depth.

Acknowledgments

The authors thank Mihir Bellare for his helpful comments.

References

1. D. Boneh and X. Boyen. Efficient selective-ID identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *Proceedings of Eurocrypt 2004*, volume 3027 of *LNCS*, pages 223–38. Springer, 2004.
2. D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *Proceedings of Eurocrypt 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, 2004.
3. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. Cryptology ePrint Archive, Report 2005/015, 2005.
4. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Proceedings of Crypto 2004*, LNCS, pages 41–55. Springer, 2004.
5. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Proceedings of Crypto 2001*, volume 2139 of *LNCS*, pages 213–29. Springer, 2001.
6. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. Cryptology ePrint Archive, Report 2005/018, 2005.
7. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *Proceedings of RSA-CT 2005*, 2005.
8. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *Proceedings of Asiacrypt 2001*, volume 2248 of *LNCS*, pages 514–32. Springer, 2001.
9. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In E. Biham, editor, *Proceedings of Eurocrypt 2003*, volume 2656 of *LNCS*. Springer, 2003.
10. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *Proceedings of Eurocrypt 2004*, volume 3027 of *LNCS*, pages 207–22. Springer, 2004.

11. Y. Dodis and N. Fazio. Public key broadcast encryption for stateless receivers. In J. Feigenbaum, editor, *Proceedings of the Digital Rights Management Workshop 2002*, volume 2696 of *LNCS*, pages 61–80. Springer, 2002.
12. Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In *Proceedings of the Workshop on Theory and Practice in Public Key Cryptography 2005*, 2005.
13. A. Fiat and M. Naor. Broadcast encryption. In D. Stinson, editor, *Proceedings of Crypto 1993*, volume 773 of *LNCS*, pages 480–91. Springer, 1993.
14. C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In Y. Zheng, editor, *Proceedings of Asiacrypt 2002*, volume 2501 of *LNCS*, pages 548–66, 2002.
15. M. Goodrich, J. Sun, and R. Tamassia. Efficient tree-based revocation in groups of low-state devices. In M. Franklin, editor, *Proceedings of Crypto 2004*, volume 3152 of *LNCS*, pages 511–27. Springer, 2004.
16. D. Halevy and A. Shamir. The LSD broadcast encryption scheme. In M. Yung, editor, *Proceedings of Crypto 2002*, volume 2442 of *LNCS*, pages 47–60, 2002.
17. J. Horwitz and B. Lynn. Towards hierarchical identity-based encryption. In L. Knudsen, editor, *Proceedings of Eurocrypt 2002*, volume 2332 of *LNCS*, pages 466–81. Springer, 2002.
18. A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *Proceedings of Algorithmic Number Theory Symposium IV*, volume 1838 of *LNCS*, pages 385–94. Springer, 2000.
19. S. Mitsunari, R. Sakai, and M. Kasahara. A new traitor tracing. *IEICE Transactions Fundamentals*, E85-A(2):481–84, 2002.
20. A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals*, E84-A(5):1234–43, 2001.
21. M. C. Mont, K. Harrison, and M. Sadler. The HP time vault service: exploiting IBE for timed release of confidential information. In *Proceedings of the International World Wide Web Conference 2003*, pages 160–69. ACM, 2003.
22. D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In J. Kilian, editor, *Proceedings of Crypto 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, 2001.
23. R. Rivest, A. Shamir, and D. Wagner. Time-lock puzzles and timed-release crypto. Technical Report MIT/LCS/TR-684, MIT Laboratory for Computer Science, 1996.
24. A. Shamir. Identity-based cryptosystems and signature schemes. In G. Blakley and D. Chaum, editors, *Proceedings of Crypto 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.
25. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *Proceedings of Eurocrypt 1997*, volume 1233 of *LNCS*, pages 256–66. Springer, 1997.
26. B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *Proceedings of Eurocrypt 2005*, LNCS. Springer, 2005.
27. D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya. ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In B. Pfitzmann, editor, *Proceedings of the ACM Conference on Computer and Communications Security 2004*, pages 354–63, 2004.