Forward-Secure and Searchable Broadcast Encryption with Short Ciphertexts and Private Keys

Nuttapong Attrapadung[1], Jun Furukawa[2], and Hideki Imai[3]

[1] Institute of Industrial Science, University of Tokyo, Japan.
nuts@imailab.iis.u-tokyo.ac.jp
[2] NEC Corporation, Japan.
j-furukawa@ay.jp.nec.com
[3] Research Center for Information Security, AIST, Japan.
h-imai@aist.go.jp

Abstract. We introduce a primitive called *Hierarchical Identity-Coupling Broadcast Encryption* (HICBE) that can be used for constructing efficient collusion-resistant public-key broadcast encryption schemes with extended properties such as forward-security and keyword-searchability. Our forward-secure broadcast encryption schemes have small ciphertext and private key sizes, in particular, independent of the number of users in the system. One of our best two constructions achieves ciphertexts of constant size and user private keys of size $O(\log^2 T)$, where T is the total number of time periods, while another achieves both ciphertexts and user private keys of size $O(\log T)$. These performances are comparable to those of the currently best *single-user* forward-secure public-key encryption scheme, while our schemes are designed for broadcasting to arbitrary sets of users. As a side result, we also formalize the notion of searchable broadcast encryption, which is a new generalization of public key encryption with keyword search. We then relate it to anonymous HICBE and present a construction with polylogarithmic performance.

1 Introduction

Broadcast encryption (BE) scheme [16] allows a broadcaster to encrypt a message to an arbitrarily designated subset S of all users in the system. Any user in S can decrypt the message by using his own private key while users outside S should not be able to do so even if all of them collude. Such a scheme is motivated by many applications such as pay-TV systems, the distribution of copyrighted materials such as CD/DVD. Public-key broadcast encryption is the one in which the broadcaster key is public. Such a scheme is typically harder to construct than private-key type ones. In what follows, we let n denote the number of all users.

The best BE scheme so far in the literature was recently proposed by Boneh, Gentry, and Waters [7]. Their scheme, which is a public-key scheme, achieves asymptotically optimal sizes, $O(1)$, for both broadcast ciphertexts and user private keys, with the price of $O(n)$-size public key. (To achieve some tradeoff, they also proposed a generalized scheme, of which one parametrization gives a scheme where both the public keys and the ciphertexts are of size $O(\sqrt{n})$). The previously best schemes [20, 19, 18], along the line of the subset-cover paradigm by Naor, Naor, and Lotspiech (NNL) [20], can only achieve a broadcast ciphertext of size $O(r)$ with each user's private key being of size $O(\log n)$, where $r = n - |S|$ is the number of revoked users. Although these schemes are improved in [3] by reducing the private key size to $O(1)$, the ciphertext is still of size $O(r)$.[4] These NNL derivatives are originally private-key schemes. Dodis and Fazio [15] gave a framework to extend these schemes to public-key versions using Hierarchical Identity-Based Encryption (HIBE) [17]. Instantiating this framework with a recent efficient HIBE scheme by Boneh, Boyen, and Goh [5] gives a public-key version of NNL-based schemes without loss in performance of ciphertext sizes.

Forward-Secure Broadcast Encryption. Unfortunately, a normal broadcast encryption scheme offers no security protection for any user whatsoever once his private key is compromised. As an extension to the normal variant in order to cope with the vulnerability against key exposure, the notion of forward security in the context of public-key broadcast encryption was first studied by Yao et al. [22]. A forward-secure public-key broadcast encryption (FS-BE) allows each user to update his private key periodically while keeping the public key unchanged. Such a scheme guarantees that even if an adversary learns the private key of some user at time period τ, messages encrypted during all time periods prior to τ remain secret. Yao et al. also proposed a FS-BE scheme achieving ciphertexts of size $O(r \log T \log n)$ while each user's private key is of size $O(\log^3 n \log T)$, where T is the maximum allowed time period. Indeed, they proposed a forward-secure HIBE scheme and then applied it to the NNL scheme in essentially the same manner as done by [15], as mentioned above. Later, Boneh et al. [5] proposed (at least two) more efficient forward-secure HIBE schemes, which when applying to the NNL scheme gives a FS-BE scheme with ciphertexts of size $O(r)$ and private keys of size $O(\log^3 n \log T)$ and another FS-BE scheme with ciphertexts of size $O(r \log T)$ and private keys of size $O((\log^2 n)(\log n + \log T))$. These schemes are the best FS-BE schemes so far in the literature.

1.1 Our Contributions.

Towards constructing a more efficient FS-BE scheme, we introduce a new primitive called *Hierarchical Identity-Coupling Broadcast Encryption* (HICBE), which can be considered as a generalization either of BE that further includes hierarchical-identity dimension together with key derivation functionality or of HIBE that further includes a user dimension together with broadcast functionality. Besides forward security, HICBE can be used to construct BE with other extended properties such as keyword-searchability, which is another feature that we study as a side result in this paper (see below).

[4] Note that one advantage of these NNL-based schemes is that, in contrast to the BGW scheme, all the other efficiency parameters, beside ciphertext sizes and private key sizes, are also of sub-linear (in n) size.

FS-BE with Short Ciphertexts and Private Keys. Using HICBE as a building block, we propose at least three new FS-BE schemes. One of our best two schemes achieves ciphertexts of size $O(1)$ and user private keys of size $O(\log^2 T)$. The other best scheme achieves ciphertexts of size $O(\log T)$ and user private keys of size $O(\log T)$. These outperform the previous schemes in terms of both overheads. In particular, they are independent of the parameters in the user dimension, namely n and r; moreover, the first scheme achieves the constant-size ciphertext. These performances of our schemes are comparable to those of the currently best single-user forward-secure public-key encryption scheme (cf. [5]). The public keys for both schemes are of size $O(n + \log T)$. Analogously to [7], we can show that this amount can be traded off to $O(\sqrt{n} + \log T)$ with ciphertext size being increased to $O(\sqrt{n})$ and $O(\sqrt{n}+\log T)$ respectively in both schemes. Security of our systems is based on the Decision Bilinear Diffie-Hellman Exponent assumption (BDHE), which is previously used in [7, 5]. We prove the security in the standard model (i.e., without random oracle).

Searchable Broadcast Encryption. Public-key BE can be applied naturally to encrypted file systems, which enable file sharing among privileged users over a public server, as already suggested in [7]. A file can be created by anyone using the public key and the privileged subset can be arbitrarily specified by the creator of the file. Due to a possible large amount of databases, a user Alice might want to retrieve only those files that contain a particular keyword of interest (among all the files in which Alice is specified as a privileged user), but without giving the server the ability to decrypt the databases. *Public-key Broadcast Encryption with Keyword Search* (BEKS) allows to do exactly this. It enables Alice to give the server a capability (or a trapdoor) to test whether a particular keyword, w, is contained in any (and only) file that includes Alice as a privileged user. This is done in such a way that (1) the server is unable to learn anything else about that file, besides the information about containment of w, and (2) all the other users outside the privileged set cannot learn anything, in particular, cannot generate such a trapdoor, even if they collude. BEKS is a new generalization of public key encryption with keyword search (PEKS) [6] that we introduce in this paper. We then relate that an anonymous ICBE (1-level HICBE) is sufficient to construct BEKS, analogously to the relation between anonymous IBE and PEKS [1].

A trivial BEKS achieving ciphertexts of size $O(n)$ can be constructed from the concatenation of PEKS-encryption of the same keyword to each privileged user. Our scheme achieves ciphertexts of size $O(r \log n)$, trapdoors of size $O(\log^3 n)$, and private keys of size $O(\log^4 n)$. Before coming up with this result, we constructively hint that even using the same technique as our FS-BE schemes (where a *non-anonymous* HICBE is sufficient), it might not be easy to construct a BEKS scheme with both ciphertext and private key of sizes independent of n. We refer for most of the results in this part to the full paper [2] due to limited space here.

2 Preliminaries

Bilinear Maps. We briefly review facts about bilinear maps. We use the standard terminology from [8]. Let $\mathbb{G}, \mathbb{G}_1$ be multiplicative groups of prime order p.

Let g be a generator of $\mathbb{G}$. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ for which the following hold: (1) e is bilinear; that is, for all $u, v \in \mathbb{G}$, $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$. (2) The map is non-degenerate: $e(g, g) \neq 1$. We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists $\mathbb{G}_1$ for which the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is efficiently computable. Although it is desirable to use asymmetric type, $e : \mathbb{G} \times \mathbb{G}' \to \mathbb{G}_1$ where $\mathbb{G} \neq \mathbb{G}'$, so that group elements will have compact representation, for simplicity we will present our schemes by the symmetric ones. Indeed, our schemes can be rephrased in terms of asymmetric maps.

Decision BDHE Assumption.[5] Let $\mathbb{G}$ be a bilinear group of prime order p. The Decision n-BDHE (Bilinear Diffie-Hellman Exponent) problem [7, 5] in $\mathbb{G}$ is stated as follows: given a vector

$$\left(g, h, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^n)}, g^{(\alpha^{n+2})}, \ldots, g^{(\alpha^{2n})}, Z\right) \in \mathbb{G}^{2n+1} \times \mathbb{G}_1$$

as input, determine whether $Z = e(g, h)^{(\alpha^{n+1})}$. We denote $g_i = g^{(\alpha^i)} \in \mathbb{G}$ for shorthand. Let $\boldsymbol{y}_{g,\alpha,n} = (g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$. An algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ has advantage ϵ in solving Decision n-BDHE in $\mathbb{G}$ if $\left|\Pr\left[\mathcal{A}\left(g, h, \boldsymbol{y}_{g,\alpha,n}, e(g_{n+1}, h)\right) = 0\right] - \Pr\left[\mathcal{A}\left(g, h, \boldsymbol{y}_{g,\alpha,n}, Z\right) = 0\right]\right| \geq \epsilon$, where the probability is over the random choice of generators $g, h \in \mathbb{G}$, the random choice of $\alpha \in \mathbb{Z}_p$, the random choice of $Z \in \mathbb{G}_1$, and the randomness of $\mathcal{A}$. We refer to the distribution on the left as $\mathcal{P}_{BDHE}$ and the distribution on the right as $\mathcal{R}_{BDHE}$. We say that the Decision (t, ϵ, n)-BDHE assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the Decision n-BDHE problem in $\mathbb{G}$. We sometimes drop t, ϵ and refer it as the Decision n-BDHE assumption in $\mathbb{G}$.

3 Hierarchical Identity-Coupling Broadcast Encryption

Model. A HICBE system consists of n users, each with index $i \in \{1, \ldots, n\}$. In usage, a user index will be "coupled" with some additional arbitrary identity tuple $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_z)$, for any I_j in some predefined identity space $\mathcal{I}$ and any $z = 1, \ldots, L$ where L is a predetermined maximum depth of tuples. The user i coupling with ID, which we will refer as a *node* (i, ID), will possess its own private key $d_{i,\mathsf{ID}}$. If $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_z)$, then for $j = 1, \ldots, z$, let $\mathsf{ID}_{|j} = (\mathrm{I}_1, \ldots, \mathrm{I}_j)$, and let $\mathsf{ID}_{|0}$ be the empty string ε. A HICBE system enables a derivation from $d_{i,\mathsf{ID}_{|z-1}}$ to $d_{i,\mathsf{ID}}$. In particular, $d_{i,(\mathrm{I}_1)}$ can be derived from d_i, the *root* private keys of i. A HICBE system enables one to encrypt a message to a set of nodes $\{(i, \mathsf{ID}) | i \in S\}$ for arbitrary $S \subseteq \{1, \ldots, n\}$, where we say that it is encrypted to *multi-node* (S, ID). If $i \in S$, the user i coupling with ID (who possesses $d_{i,\mathsf{ID}}$) can decrypt this ciphertext. When $L = 1$, we simply call it an ICBE.

[5] This holds in the generic bilinear group model with the computational lower bound of $\Omega(\sqrt{p/n})$ on the difficulty of breaking (cf.[5]). Cheon [14] recently showed a concrete attack with roughly the same complexity. It is recommended to either increase p (to $\approx$ 220-bit size for $n = 2^{64}$ to achieve 2^{80} security) or use p of a special form where $p - 1$ and $p + 1$ have no small divisor greater than $\log^2 p$ to avoid the attack.

Formally, a HICBE system is made up of five randomized algorithms as follows. For simplicity, we define it as a key encapsulation mechanism (KEM).

$\mathsf{Setup}(n, L)$**:** Takes as input the number of all users n and the maximum depth L of the identity hierarchy. It outputs a public key pk and a master key msk.

$\mathsf{PrivKeyGen}(i, \mathsf{pk}, \mathsf{msk})$**:** Takes as input a user index i, the public key pk, and the master key msk. It outputs a root private key d_i of user i.

$\mathsf{Derive}(\mathsf{pk}, i, \mathsf{ID}, d_{i,\mathsf{ID}_{|z-1}})$**:** Takes as input the public key pk, a user index i, an identity ID of depth z, and the private key $d_{i,\mathsf{ID}_{|z-1}}$ of user i coupling with the parent identity $\mathsf{ID}_{|z-1}$. It outputs $d_{i,\mathsf{ID}}$. Here $d_{i,\mathsf{ID}_{|0}} = d_i$.

$\mathsf{Encrypt}(\mathsf{pk}, S, \mathsf{ID})$**:** Takes as input the public key pk, a subset $S \subseteq \{1, \ldots, n\}$, and an identity tuple ID. It outputs a pair (hdr, K) where hdr is called the header and $K \in \mathcal{K}$ is a message encryption key. We will also refer to hdr as the broadcast ciphertext.

$\mathsf{Decrypt}(\mathsf{pk}, S, i, d_{i,\mathsf{ID}}, \mathsf{hdr})$**:** Takes as input the pk, a subset S, a user i, the private key $d_{i,\mathsf{ID}}$ of user i coupling with ID, and the header hdr. If $i \in S$ it outputs $K \in \mathcal{K}$ else outputs a special symbol '$\notin$'.

The correctness consistency can be defined straightforwardly and is omitted here.

Confidentiality. We define semantic security of HICBE by the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$; both are given n, L as input.

Setup. The challenger $\mathcal{C}$ runs $\mathsf{Setup}(n, L)$ to obtain a public key pk and the master key msk. It then gives the public key pk to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_\mu$ where each is one of two types:

- Private key query $\langle i, \mathsf{ID}\rangle$. $\mathcal{C}$ responds by running algorithm $\mathsf{PrivKeyGen}$ and Derive to derive the private key $d_{i,\mathsf{ID}}$, corresponding to the node (i, ID), then sends $d_{i,\mathsf{ID}}$ to $\mathcal{A}$.
- Decryption query $\langle S, \mathsf{ID}, i, \mathsf{hdr}\rangle$ where $i \in S$. $\mathcal{C}$ responds by running algorithm $\mathsf{PrivKeyGen}$ and Derive to derive the private key $d_{i,\mathsf{ID}}$, corresponding to the node (i, ID). It then gives to $\mathcal{A}$ the output from $\mathsf{Decrypt}(\mathsf{pk}, S, i, d_{i,\mathsf{ID}}, \mathsf{hdr})$.

Challenge. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs $(S^\star, \mathsf{ID}^\star)$ which is the multi-node it wants to attack, where $S^\star \subseteq \{1, \ldots, n\}$. The only restriction is that $\mathcal{A}$ did not previously issue a private key query for $\langle i, \mathsf{ID}\rangle$ such that $i \in S^\star$ and that either $\mathsf{ID} = \mathsf{ID}^\star$ or ID is a prefix of $\mathsf{ID}^\star$. $\mathcal{C}$ then compute $(\mathsf{hdr}^\star, K) \xleftarrow{R} \mathsf{Encrypt}(\mathsf{pk}, S^\star, \mathsf{ID}^\star)$ where $K \in \mathcal{K}$. Next $\mathcal{C}$ picks a random $b \in \{0, 1\}$. It sets $K_b = K$ and picks a random K_{1-b} in $\mathcal{K}$. It then gives $(\mathsf{hdr}^\star, K_0, K_1)$ to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ issues additional queries $q_{\mu+1}, \ldots, q_\nu$ where each is one of two types:

- Private key query $\langle i, \mathsf{ID}\rangle$ such that if $i \in S^\star$ then neither $\mathsf{ID} = \mathsf{ID}^\star$ nor ID is a prefix of $\mathsf{ID}^\star$, else $(i \notin S^\star)$ ID can be arbitrary.
- Decryption query $\langle S, \mathsf{ID}, i, \mathsf{hdr}\rangle$ where $i \in S$ and $S \subseteq S^\star$.[6] The only constraint is that $\mathsf{hdr} \neq \mathsf{hdr}^\star$ if either $\mathsf{ID} = \mathsf{ID}^\star$ or ID is a prefix of $\mathsf{ID}^\star$.

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

---

[6] It is WLOG that we just restrict $S \subseteq S^\star$ since for S such that $S \not\subseteq S^\star$, one can make a private key query for some $i \in S \setminus S^\star$ and perform the decryption oneself.

Guess Finally $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ for b and wins the game if $b = b'$.

We refer to such an adversary $\mathcal{A}$ as an IND-aID-aSet-CCA adversary and the above game as the IND-aID-aSet-CCA game. Weaker notions of security can be defined by modifying the above game so that it is required that the adversary must commit ahead of time to the target subset $S^\star$ or the target identity $\mathsf{ID}^\star$ or both. These notions are analogous to the notion of selective-identity secure HIBE, defined in [12, 13]. We have 4 possible combinations: the game IND-xID-ySet-CCA where $(\mathsf{x}, \mathsf{y}) \in \{(\mathsf{a}, \mathsf{a}), (\mathsf{a}, \mathsf{s}), (\mathsf{s}, \mathsf{a}), (\mathsf{s}, \mathsf{s})\}$. If $(\mathsf{x}, \mathsf{y}) = (\mathsf{s}, *)$ then it is exactly the same as IND-aID-aSet-CCA except that $\mathcal{A}$ must disclose to $\mathcal{C}$ the target identity $\mathsf{ID}^\star$ before the Setup phase. Analogously, if $(\mathsf{x}, \mathsf{y}) = (*, \mathsf{s})$, $\mathcal{A}$ must disclose the target subset $S^\star$ before the Setup phase. For only the case of (s, s), it is further required that the restrictions on private key queries from phase 2 also hold in phase 1. Intuitively, s means selective while a means adaptive security.

We define the advantage of the adversary $\mathcal{A}$ in attacking the HICBE scheme $\mathcal{E}$ in the game IND-xID-ySet-CCA as $\mathsf{AdvHICBE}_{\mathsf{xy}}(\mathcal{E}, \mathcal{A}) = |\Pr[b = b'] - \frac{1}{2}|$, where the probability is over the random bits used by $\mathcal{C}$ and $\mathcal{A}$ in that game.

Definition 1. *We say that a HICBE system $\mathcal{E}$ is $(t, q_{\mathsf{P}}, q_{\mathsf{D}}, \epsilon)$-IND-xID-ySet-CCA-secure if for any t-time IND-xID-ySet-CCA adversary $\mathcal{A}$ that makes at most q_{P} chosen private key queries and at most q_{D} chosen decryption queries, we have that $\mathsf{AdvHICBE}_{\mathsf{xy}}(\mathcal{E}, \mathcal{A}) < \epsilon$. We say that a HICBE system $\mathcal{E}$ is $(t, q_{\mathsf{P}}, \epsilon)$-IND-xID-ySet-CPA-secure if $\mathcal{E}$ is $(t, q_{\mathsf{P}}, 0, \epsilon)$-IND-xID-ySet-CCA-secure.*

Anonymity. Recipient anonymity is the property that the adversary be unable to distinguish the ciphertext intended for a chosen identity from another one intended for a random identity. We capture such a property via what we name ANO-xID-ySet-CCA[Δ] notion, where $\Delta \subseteq \{0, \dots, L\}$ indicates a set of levels that satisfy anonymity, with 0 corresponds to the anonymity of the set S. This is a generalized notion from [1]. We refer to the full paper [2] for the details .

4 HICBE Constructions

In this section, we give our first two HICBE constructions. A HICBE system must have both broadcast and hierarchical-identity-based derivation properties. To achieve this we will combine some techniques from the BGW broadcast encryption [7] with the BB and BBG HIBE systems by Boneh-Boyen [4] and Boneh-Boyen-Goh [5] respectively. The reader is encouraged to refer to the full paper [2] for the intuition into the design.

4.1 Our First HICBE Construction Based on BGW and BB

We first show how to combine the basic BGW scheme with the BB HIBE scheme. We assume that the identity space $\mathcal{I}$ is $\mathbb{Z}_p$. Thus, if ID is of depth z then $\mathsf{ID} = (\mathrm{I}_1, \dots, \mathrm{I}_z) \in \mathbb{Z}_p{}^z$. As in [4], we can later extend the construction to arbitrary identities in $\{0,1\}^*$ by first hashing each I_j using a collision resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$. We follow almost the same terminology from [7, 4]. This scheme, denoted by BasicHICBE1, works as follows.

Setup(n, L)**:** Let $\mathbb{G}$ be a bilinear group of prime order p. It first picks a random generator $g \in \mathbb{G}$ and a random $\alpha \in \mathbb{Z}_p$. It computes $g_i = g^{(\alpha^i)} \in \mathbb{G}$ for $i = 1, 2, \ldots, n, n+2, \ldots, 2n$. Next, it picks a random $\gamma \in \mathbb{Z}_p$ and sets $v = g^\gamma \in \mathbb{G}$. It then picks random elements $h_1, \ldots, h_L \in \mathbb{G}$. The public key is:

$$\mathsf{pk} = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v, h_1, \ldots, h_L) \in \mathbb{G}^{2n+L+1}.$$

The master key is $\mathsf{msk} = \gamma$. For $j = 1, \ldots, L$, we define $F_j : \mathbb{Z}_p \to \mathbb{G}$ to be the function: $F_j(x) = g_1^x h_j$. The algorithm outputs pk and msk.

PrivKeyGen$(i, \mathsf{pk}, \mathsf{msk})$**:** Set a root private key for i as $d_i = (g_i)^\gamma = v^{(\alpha^i)} \in \mathbb{G}$.

Derive$(\mathsf{pk}, i, \mathsf{ID}, d_{i,\mathsf{ID}_{|z-1}})$**:** To generate the private key for node (i, ID) where $i \in \{1, \ldots, n\}$ and $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_z) \in {\mathbb{Z}_p}^z$ of depth $z \le L$, pick random elements $s_1, \ldots, s_z \in \mathbb{Z}_p$ and output

$$d_{i,\mathsf{ID}} = \left((g_i)^\gamma \cdot \prod_{j=1}^{z} F_j(\mathrm{I}_j)^{s_j} \ , \ g^{s_1} \ , \ \ldots \ , \ g^{s_z} \right) \in \mathbb{G}^{z+1}.$$

Note that the private key for node (i, ID) can be generated just given a private key for node $(i, \mathsf{ID}_{|z-1})$ where $\mathsf{ID}_{|z-1} = (\mathrm{I}_1, \ldots \mathrm{I}_{z-1}) \in {\mathbb{Z}_p}^{z-1}$, as required. Indeed, let $d_{i,\mathsf{ID}_{|z-1}} = (a_0, \ldots, a_{z-1})$ be the private key for node $(i, \mathsf{ID}_{|z-1})$. To generate $d_{i,\mathsf{ID}}$, pick a random $s_z \in \mathbb{Z}_p$ and output $d_{i,\mathsf{ID}} = (a_0 \cdot F_z(\mathrm{I}_z)^{s_z}, a_1, \ldots, a_{z-1}, g^{s_z})$.

Encrypt$(\mathsf{pk}, S, \mathsf{ID})$**:** Pick a random $t \in \mathbb{Z}_p$ and set $K = e(g_{n+1}, g)^t$. The value $e(g_{n+1}, g)$ can be computed as $e(g_n, g_1)$. Let $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_z)$. It outputs (hdr, K) where we let

$$\mathsf{hdr} = \left(g^t, \ (v \cdot \prod_{j \in S} g_{n+1-j})^t \ , \ F_1(\mathrm{I}_1)^t \ , \ \ldots \ , \ F_z(\mathrm{I}_z)^t \right) \in \mathbb{G}^{z+2}.$$

Decrypt$(\mathsf{pk}, S, i, d_{i,\mathsf{ID}}, \mathsf{hdr})$**:** Parse the header as $\mathsf{hdr} = (C_0, C_1, A_1, \ldots, A_z) \in \mathbb{G}^{z+2}$. Also parse $d_{i,\mathsf{ID}} = (a_0, \ldots, a_z) \in \mathbb{G}^{z+1}$. Then output

$$K = e(g_i, C_1) \cdot \prod_{j=1}^{z} e(A_j, a_j) \ / \ e(a_0 \cdot \prod_{\substack{j \in S \\ j \ne i}} g_{n+1-j+i}, \ C_0).$$

The correctness verification is straightforward. The scheme inherits a good property of the BGW scheme: the ciphertext size and user private key size are independent of n. Indeed, when we let $\mathsf{ID} = \varepsilon$, the corresponding algorithms become those of the basic BGW scheme.

Theorem 1. *Let $\mathbb{G}$ be a bilinear group of prime order p. Suppose the Decision (t, ϵ, n)-BDHE assumption holds in $\mathbb{G}$. Then the BasicHICBE1 system for n users and maximum depth L is $(t', q_{\mathsf{P}}, \epsilon)$-IND-sID-sSet-CPA-secure for any n, L, q_{P}, and $t' < t - \Theta(\tau_{\mathsf{exp}} L q_{\mathsf{P}})$ where τ_{exp} is the maximum time for an exponentiation in $\mathbb{G}$.*

The security proof, although vaguely resembles those of BGW and BB, is not straightforward as we have to simulate both sub-systems simultaneously. Intuitively, the implicit "orthogonality" of BGW and BB allows us to prove the security of the combined scheme. We omit it here (and refer to [2]) and will focus on a similar but somewhat more interesting proof of the second scheme.

4.2 Our Second HICBE Construction Based on BGW and BBG

Our method of integrating the BGW system can also be applied to the BBG HIBE scheme analogously to the previous integration. In contrast, this time we achieve a feature of "reusing" the public key from the BGW portion to be used for the BBG portion. Consequently, the resulting scheme has exactly the same public key as the BGW scheme except for only one additional element of $\mathbb{G}$.

We will assume that $L \le n$, otherwise just create dummy users so as to be so; a more efficient way will be discussed in the next subsection. As usual we can assume that $\mathcal{I}$ is $\mathbb{Z}_p$. The scheme, denoted by $\mathsf{BasicHICBE2}$, works as follows.

$\mathsf{Setup}(n, L)$**:** The algorithm first picks a random generator $g \in \mathbb{G}$ and a random $\alpha \in \mathbb{Z}_p$. It computes $g_i = g^{(\alpha^i)} \in \mathbb{G}$ for $i = 1, 2, \ldots, n, n+2, \ldots, 2n$. Next, it randomly picks $y \in \mathbb{G}$, $\gamma \in \mathbb{Z}_p$ and sets $v = g^\gamma \in \mathbb{G}$. The public key is:

$$\mathsf{pk} = (g, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n}, v, y) \in \mathbb{G}^{2n+2}.$$

The master key is $\mathsf{msk} = \gamma$. It outputs $(\mathsf{pk}, \mathsf{msk})$. For conceptual purpose, let $h_j = g_{n+1-j}$ for $j = 1, \ldots, L$; intuitively, the h_j terms will be used to visually indicate the BBG portion, while the g_j terms are for the BGW portion.

$\mathsf{PrivKeyGen}(i, \mathsf{pk}, \mathsf{msk})$**:** Set a root private key for i as $d_i = (g_i)^\gamma = v^{(\alpha^i)} \in \mathbb{G}$.

$\mathsf{Derive}(\mathsf{pk}, i, \mathsf{ID}, d_{i,\mathsf{ID}_{|z-1}})$**:** To generate the private key for node (i, ID) where $i \in \{1, \ldots, n\}$ and $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_z) \in {\mathbb{Z}_p}^z$ of depth $z \le L$, pick a random element $s \in \mathbb{Z}_p$ and output

$$d_{i,\mathsf{ID}} = \Big((g_i)^\gamma \cdot (h_1^{\mathrm{I}_1} \cdots h_z^{\mathrm{I}_z} \cdot y)^s \ , \ g^s \ , \ h_{z+1}^s \ , \ \ldots \ , \ h_L^s \Big) \in \mathbb{G}^{2+L-z}.$$

Note that the private key for node (i, ID) can be generated just given a private key for node $(i, \mathsf{ID}_{|z-1})$ where $\mathsf{ID}_{|z-1} = (\mathrm{I}_1, \ldots \mathrm{I}_{z-1}) \in {\mathbb{Z}_p}^{z-1}$, as required. Indeed, let $d_{i,\mathsf{ID}_{|z-1}} = (a_0, a_1, b_z, \ldots, b_L)$ be the private key for node $(i, \mathsf{ID}_{|z-1})$. To generate $d_{i,\mathsf{ID}}$, pick a random $\delta \in \mathbb{Z}_p$ and output $d_{i,\mathsf{ID}} = \Big(a_0 \cdot b_z^{\mathrm{I}_z} \cdot (h_1^{\mathrm{I}_1} \cdots h_z^{\mathrm{I}_z} \cdot y)^\delta \ , \ a_1 \cdot g^\delta \ , \ b_{z+1} \cdot h_{z+1}^\delta \ , \ \ldots \ , \ b_L \cdot h_L^\delta \Big)$. This key has a proper distribution as a private key for node (i, ID) with the randomness $s = s' + \delta \in \mathbb{Z}_p$, where s' is the randomness in $d_{i,\mathsf{ID}_{|z-1}}$. Note that the private key $d_{i,\mathsf{ID}}$ becomes shorter as the depth of ID increases.

$\mathsf{Encrypt}(\mathsf{pk}, S, \mathsf{ID})$**:** Pick a random $t \in \mathbb{Z}_p$ and set $K = e(g_{n+1}, g)^t$. The value $e(g_{n+1}, g)$ can be computed as $e(g_n, g_1)$. Let $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_z)$. It outputs (hdr, K) where we let

$$\mathsf{hdr} = \left(g^t, \ (v \cdot \prod_{j \in S} g_{n+1-j})^t \ , \ (h_1^{\mathrm{I}_1} \cdots h_z^{\mathrm{I}_z} \cdot y)^t \right) \in \mathbb{G}^3.$$

$\mathsf{Decrypt}(\mathsf{pk}, S, i, d_{i,\mathsf{ID}}, \mathsf{hdr})$**:** Let $\mathsf{hdr} = (C_0, C_1, C_2) \in \mathbb{G}^3$ and let $d_{i,\mathsf{ID}} = (a_0, a_1, b_{z+1}, \ldots, b_L) \in \mathbb{G}^{2+L-z}$. Then output

$$K = e(g_i, C_1) \cdot e(C_2, a_1) \ / \ e(a_0 \cdot \prod_{\substack{j \in S \\ j \neq i}} g_{n+1-j+i}, \ C_0).$$

The scheme inherits good properties from both the BGW scheme: the ciphertext size and user private key size are independent of n, and the BBG scheme: the ciphertext size is constant. One difference from the BBG system is that we let the h_j terms be of special forms, namely $h_j = g_{n+1-j}$, instead of random elements in $\mathbb{G}$ as in [5]. This allows us to save the public key size since those g_j terms are already used for the BGW system. Indeed, suppose that the BGW BE system has been already established, it can be augmented to a HICBE version by just once publishing one random element, namely $y \in \mathbb{G}$, as an additional public key. Note that defining h_j terms in this way is also crucial to the security proof. We prove the security under the Decision n-BDHE assumption. This strong assumption is already necessary for both the (stand-alone) BGW and BBG systems.[7]

Theorem 2. *Let $\mathbb{G}$ be a bilinear group of prime order p. Suppose the Decision (t, ϵ, n)-BDHE assumption holds in $\mathbb{G}$. Then the BasicHICBE2 scheme for n users and maximum depth L is $(t', q_{\mathsf{P}}, \epsilon)$-IND-sID-sSet-CPA-secure for arbitrary n, L such that $L \leq n$ and q_{P}, and any $t' < t - \Theta(\tau_{\mathsf{exp}} L q_{\mathsf{P}})$ where τ_{exp} is the maximum time for an exponentiation in $\mathbb{G}$.*

Proof. Suppose there exists an adversary, $\mathcal{A}$, that has advantage ϵ in attacking the HICBE scheme. We build an algorithm $\mathcal{B}$ that solves the Decision n-BDHE problem in $\mathbb{G}$. $\mathcal{B}$ is given as input a random n-BDHE challenge $(g, h, \boldsymbol{y}_{g,\alpha,n}, Z)$, where $\boldsymbol{y}_{g,\alpha,n} = (g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$ and Z is either $e(g_{n+1}, h)$ or a random element in $\mathbb{G}_1$ (recall that $g_j = g^{(\alpha^j)}$). Algorithm $\mathcal{B}$ proceeds as follows.

Initialization. The selective (identity, subset) game begins with $\mathcal{A}$ first outputting a multi-node $(S^\star, \mathsf{ID}^\star)$ where $S^\star \subseteq \{1, \ldots, n\}$ and $\mathsf{ID}^\star = (\mathrm{I}_1^\star, \ldots, \mathrm{I}_z^\star) \in {\mathbb{Z}_p}^z$ of depth $z \leq L$ that it intends to attack.

Setup. To generate pk, algorithm $\mathcal{B}$ randomly chooses $u, \sigma \in \mathbb{Z}_p$ and sets

$$v = g^u \cdot (\prod_{j \in S^\star} g_{n+1-j})^{-1}, \quad y = g^\sigma \cdot \prod_{j=1}^{z} g_{n+1-j}^{-\mathrm{I}_j^\star}.$$

It gives $\mathcal{A}$ the $\mathsf{pk} = (g, \boldsymbol{y}_{g,\alpha,n}, v, y)$. Since g, α, u, σ are chosen randomly and independently, pk has an identical distribution to that in the actual construction.

Phase 1. $\mathcal{A}$ issues up to q_{P} private key queries. Consider a query for the private key corresponding to node (i, ID), of which $\mathsf{ID} = (\mathrm{I}_1, \ldots, \mathrm{I}_w) \in {\mathbb{Z}_p}^w$ where $w \leq L$. We distinguish two cases according to whether i is in $S^\star$ or not.

If $i \notin S^\star$ then $\mathcal{B}$ responds to the query by first computing a root private key d_i from which it can then construct a private key $d_{i,\mathsf{ID}}$ for the request node (i, ID). In this case, $\mathcal{B}$ computes d_i as $d_i = g_i^u \cdot (\prod_{j \in S^\star} g_{n+1-j+i})^{-1}$. Indeed, we have $d_i = (g^u (\prod_{j \in S^\star} g_{n+1-j})^{-1})^{(\alpha^i)} = v^{(\alpha^i)}$, as required.

If $i \in S^\star$ then from the restriction of the private key query, it must be that ID is neither $\mathsf{ID}^\star$ nor any prefix of $\mathsf{ID}^\star$. We further distinguish two cases according to whether $\mathsf{ID}^\star$ is a prefix of ID or not.

7 It was later shown in [5, full] that a *truncated* form of Decision n-BDHE, namely the Decision n-wBDHI*, indeed suffices for BBG. This assumption is defined exactly the same as the former except that we change the vector $\boldsymbol{y}_{g,\alpha,n}$ to $\boldsymbol{y}^*_{g,\alpha,n} := (g_1, \ldots, g_n)$.

Case 1: $\mathsf{ID}^\star$ is *not* a prefix of ID. Then there must exist $k \le z$ such that it is the smallest index satisfying $\mathrm{I}_k \neq \mathrm{I}_k^\star$. $\mathcal{B}$ responds to the query by first computing a private key for node $(i, \mathsf{ID}_{|k})$ from which it then constructs a private key for the request node (i, ID). $\mathcal{B}$ picks random elements $s \in \mathbb{Z}_p$. We pose $\tilde{s} = s + \alpha^k/(\mathrm{I}_k - \mathrm{I}_k^\star)$. Note that $\tilde{s}$ is unknown to $\mathcal{B}$. Next, $\mathcal{B}$ generates the private key

$$(a_0, a_1, b_{k+1}, \ldots, b_L) = \Big(v^{(\alpha^i)} \cdot (h_1^{\mathrm{I}_1} \cdots h_k^{\mathrm{I}_k} \cdot y)^{\tilde{s}} \ , \ g^{\tilde{s}} \ , \ h_{k+1}^{\tilde{s}} \ , \ \ldots \ , \ h_L^{\tilde{s}} \Big) \tag{1}$$

which is a valid random private key for node $(i, \mathsf{ID}_{|k})$ by definition. We show that $\mathcal{B}$ can compute all elements of this private key given the values that it knows. Recall that $h_j = g_{n+1-j}$. To generate a_0, we first assume that $k < z$, and observe

$$\begin{aligned} a_0 &= g_i^u \Big(\prod_{j \in S^\star} g_{n+1-j+i} \Big)^{-1} \cdot (g^\sigma \cdot \underbrace{\prod_{j=1}^{k-1} g_{n+1-j}^{\mathrm{I}_j - \mathrm{I}_j^\star}}_{=1} \cdot g_{n+1-k}^{\mathrm{I}_k - \mathrm{I}_k^\star} \cdot \prod_{j=k+1}^{z} g_{n+1-j}^{-\mathrm{I}_j^\star})^{\tilde{s}} \\ &= g_i^u \Big(\prod_{\substack{j \in S^\star \\ j \neq i}} g_{n+1-j+i} \Big)^{-1} \cdot \underbrace{g_{n+1}^{-1} \cdot g_{n+1-k}^{(\mathrm{I}_k - \mathrm{I}_k^\star)\tilde{s}}}_{T_1} \cdot \underbrace{g^{\sigma \tilde{s}}}_{T_2} \cdot \underbrace{\prod_{j=k+1}^{z} g_{n+1-j}^{-\mathrm{I}_j^\star \tilde{s}}}_{T_3} . \end{aligned}$$

The term T_1 can be computed by $\mathcal{B}$ since

$$T_1 = g_{n+1}^{-1} \cdot g_{n+1-k}^{(\mathrm{I}_k - \mathrm{I}_k^\star)(s + \frac{\alpha^k}{\mathrm{I}_k - \mathrm{I}_k^\star})} = g_{n+1}^{-1} \cdot g_{n+1-k}^{(\mathrm{I}_k - \mathrm{I}_k^\star)s} \cdot g_{n+1-k}^{\alpha^k} = g_{n+1-k}^{(\mathrm{I}_k - \mathrm{I}_k^\star)s},$$

where the unknown term g_{n+1} is canceled out. The term T_2 can be computed by using g_k, which is not g_{n+1} since $k \le z \le L \le n$. Each term in the product T_3 is computable since $g_{n+1-j}^{\tilde{s}} = g_{n+1-j}^{s} \cdot g_{n+1-j+k}^{1/(\mathrm{I}_k - \mathrm{I}_k^\star)}$ and for $j = k+1, \ldots, z$, the terms $g_{n+1-j}, g_{n+1-j+k}$ are not equal to g_{n+1} hence can be computed. It is left to consider the case $k = z$. In this case, a_0 is exactly the same as above except that the last product term, i.e., T_3, does not appear. The analysis of computability by $\mathcal{B}$ thus follows from the same argument.

The component a_1 can be generated since $a_1 = g^{\tilde{s}} = g^s \cdot g_k^{1/(\mathrm{I}_k - \mathrm{I}_k^\star)}$. For $j = k+1, \ldots, L$, the value b_j can be computed as $b_j = h_j^{\tilde{s}} = h_j^s \cdot g_{n+1-j+k}^{1/(\mathrm{I}_k - \mathrm{I}_k^\star)}$.

Case 2: $\mathsf{ID}^\star$ is a prefix of ID. Then it holds that $z+1 \le w$. $\mathcal{B}$ responds to the query by first computing a private key for node $(i, \mathsf{ID}_{|z+1})$ from which it then construct a private key for the request node (i, ID). $\mathcal{B}$ picks random elements $s \in \mathbb{Z}_p$. We pose $\tilde{s} = s + \alpha^{z+1}/\mathrm{I}_{z+1}$. Note that $\tilde{s}$ is unknown to $\mathcal{B}$. Next, $\mathcal{B}$ generates the private key in exactly the same form as Eq.(1) (change k to $z+1$, of course). From a similar observation as above, one can show that $\mathcal{B}$ can compute this key.

Challenge. To generate the challenge, $\mathcal{B}$ computes $\mathsf{hdr}^\star$ as (h, h^u, h^σ). It then randomly chooses a bit $b \in \{0,1\}$ and sets $K_b = Z$ and picks a random K_{1-b} in $\mathbb{G}_1$. $\mathcal{B}$ then gives $(\mathsf{hdr}^\star, K_0, K_1)$ to $\mathcal{A}$.

We claim that when $Z = e(g_{n+1}, h)$ (that is, the input to $\mathcal{B}$ is a n-BDHE tuple) then $(\mathsf{hdr}^\star, K_0, K_1)$ is a valid challenge to $\mathcal{A}$ as in a real attack game. To see this, write $h = g^t$ for some (unknown) $t \in \mathbb{Z}_p$. Then, we have that

$$h^u = (g^u)^t = (g^u(\prod_{j \in S^\star} g_{n+1-j})^{-1}(\prod_{j \in S^\star} g_{n+1-j}))^t = (v \prod_{j \in S^\star} g_{n+1-j})^t,$$

$$h^\sigma = \Big(\prod_{j=1}^{z} g_{n+1-j}^{\mathrm{I}_j^\star} \cdot (g^\sigma \cdot \prod_{j=1}^{z} g_{n+1-j}^{-\mathrm{I}_j^\star})\Big)^t = (h_1^{\mathrm{I}_1^\star} \cdots h_z^{\mathrm{I}_z^\star} \cdot y)^t.$$

Thus, by definition, (h, h^u, h^σ) is a valid encryption of the key $e(g_{n+1}, g)^t$. Also, $e(g_{n+1}, g)^t = e(g_{n+1}, h) = Z = K_b$ and hence $(\mathsf{hdr}^\star, K_0, K_1)$ is a valid challenge.

On the other hand, when Z is random in $\mathbb{G}_1$ (that is, the input to $\mathcal{B}$ is a random tuple) then K_0, K_1 are just random independent elements of $\mathbb{G}_1$.

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

Guess. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. If $b = b'$ then $\mathcal{B}$ outputs 1 (meaning $Z = e(g_{n+1}, h)$). Otherwise, it outputs 0 (meaning Z is random in $\mathbb{G}_1$).

We see that if $(g, h, \boldsymbol{y}_{g,\alpha,n}, Z)$ is sampled from $\mathcal{R}_{BDHE}$ then $\Pr[\mathcal{B}(g, h, \boldsymbol{y}_{g,\alpha,n}, Z) = 0] = \frac{1}{2}$. On the other hand, if $(g, h, \boldsymbol{y}_{g,\alpha,n}, Z)$ is sampled from $\mathcal{P}_{BDHE}$ then $|\Pr[\mathcal{B}(g, h, \boldsymbol{y}_{g,\alpha,n}, Z) = 0] - \frac{1}{2}| \geq \epsilon$. It follows that $\mathcal{B}$ has advantage at least ϵ in solving n-BDHE problem in $\mathbb{G}$. This concludes the proof of Theorem 2. $\square$

4.3 Extensions

Modification. Recall that for $\mathsf{BasicHICBE2}$ when $L > n$, we created dummy users so that the effective number of users is L. The resulting pk contained $2L+2$ elements of $\mathbb{G}$. We now give a more efficient scheme in this case ($L > n$). First, we change 'n' in all appearances in the description of $\mathsf{BasicHICBE2}$ to 'L' except that the user indexes are as usual: $\{1, \ldots, n\}$. Then we modify the public key to $\mathsf{pk} = (g, g_1, \ldots, g_L, g_{L+2}, \ldots, g_{L+n}, v, y) \in \mathbb{G}^{L+n+2}$, which is of smaller size than that of the above method. This modified scheme is secure under the Decision L-BDHE assumption. However, it can be shown to be secure under a weaker one which is a new assumption that we call Decision $\langle L, n\rangle$-BDHE. (Two values are specified instead of only one). It is defined exactly the same as the Decision L-BDHE except that we change $\boldsymbol{y}_{g,\alpha,L}$ to $\boldsymbol{y}_{g,\alpha,\langle L,n\rangle} := (g_1, \ldots, g_L, g_{L+2}, \ldots, g_{L+n})$.

Generalizations. Without going into details, we can also combine the BGW system with the Hybrid $\mathsf{BB/BBG}$ scheme [5, full §4.2], which can trade off the public key and private key sizes with the ciphertext size. We denote this scheme by $\mathsf{BasicHICBE}(\omega)$ for parameter $\omega \in [0,1]$. It becomes $\mathsf{BasicHICBE1}$ when $\omega = 1$ and $\mathsf{BasicHICBE2}$ when $\omega = 0$. In this scheme, the public key, the private key, and the ciphertext contains $L^\omega + \max(L^{1-\omega}, n) + n + 1$, $\leq L^{1-\omega} + L^\omega + 1$, and $\leq L^\omega + 2$ elements in $\mathbb{G}$ respectively. It can also be further generalized in the other dimension, namely the user dimension, in the same manner as the generalized BGW scheme [7], which can trade off the public key size with the ciphertext size while the private key size remains fixed. In the resulting scheme, denoted by $\mathsf{GenHICBE}(\omega, \mu)$, for $\mu \in [0,1]$, the public key, the private key, and the ciphertext contains $L^\omega + \max(L^{1-\omega}, n^\mu) + n^\mu + n^{1-\mu}$, $\leq L^{1-\omega} + L^\omega + 1$, $\leq L^\omega + n^{1-\mu} + 1$ elements in $\mathbb{G}$ respectively. Note that it becomes $\mathsf{BasicHICBE}(\omega)$ when $\mu = 1$.

Chosen-Ciphertext and Adaptive-ID Security. We use the conversion due to Canetti et al. [13] or its derivatives [9, 10] (adapted to the case of HICBE appropriately) to obtain $\mathsf{IND\text{-}sID\text{-}sSet\text{-}CCA}$-secure schemes. An $\mathsf{IND\text{-}aID\text{-}sSet\text{-}CCA}$-secure scheme can be constructed by combining the BGW system with Waters' HIBE [21] in essentially the same way as our previous two schemes.

5 Forward-Secure Public-key Broadcast Encryption

Model for FS-BE. The syntax of a forward-secure public-key broadcast encryption (FS-BE) scheme is introduced in [22]. Following [7], for simplicity we define it as a KEM. A key-evolving broadcast encryption is made up of six randomized algorithms. Via $(\mathsf{pk}, \mathsf{msk}_0) \xleftarrow{R} \mathsf{Setup}(n, T)$, where n is the number of receivers and T is the total number of time periods, the setup algorithm produces a public key pk and an initial master private key msk_0; via $\mathsf{msk}_{i,\tau} \xleftarrow{R} \mathsf{MasUpdate}(\mathsf{pk}, \tau, \mathsf{msk}_{\tau-1})$ the master key update algorithm outputs a new private key $\mathsf{msk}_{i,\tau}$ of user i for time period τ; via $\mathsf{sk}_{i,\tau} \xleftarrow{R} \mathsf{Regist}(i, \tau, \mathsf{pk}, \mathsf{msk}_\tau)$ the center outputs a private key $\mathsf{sk}_{i,\tau}$ of user i for time period τ; via $\mathsf{sk}_{i,\tau} \xleftarrow{R} \mathsf{Update}(\mathsf{pk}, i, \tau, \mathsf{sk}_{i,\tau-1})$ the user i updates his private key to $\mathsf{sk}_{i,\tau}$ for the consecutive time period; via $(\mathsf{hdr}, K) \xleftarrow{R} \mathsf{Encrypt}(\mathsf{pk}, S, \tau)$, where S is the set of recipients, a sender outputs a pair (hdr, K), a header and a message encryption key; via $K \xleftarrow{R} \mathsf{Decrypt}(\mathsf{pk}, S, i, \mathsf{sk}_{i,\tau}, \mathsf{hdr})$ a recipient $i \in S$ outputs $K \in \mathcal{K}$. A scheme is correct if (1) when $\mathsf{pk}, \mathsf{msk}_\tau, \mathsf{sk}_{i,\tau-1}$ are correctly generated, the distributions of private keys output from $\mathsf{Regist}(i, \tau, \mathsf{pk}, \mathsf{msk}_\tau)$ and from $\mathsf{Update}(\mathsf{pk}, i, \tau, \mathsf{sk}_{i,\tau-1})$ are the same; (2) $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$ are consistent (in the standard way).

Security Notions. We define semantic security of a key-evolving BE in essentially the same way as in the case of HICBE system. Such a notion is introduced by Yao et al. [22]. We reformalize and briefly state it here. (See the full paper [2] for details). We define eight combinations of notions called $\mathsf{IND\text{-}xFS}_i\mathsf{\text{-}ySet\text{-}CCA}$ security where $(\mathsf{x}, \mathsf{y}) \in \{(\mathsf{a}, \mathsf{a}), (\mathsf{a}, \mathsf{s}), (\mathsf{s}, \mathsf{a}), (\mathsf{s}, \mathsf{s})\}$, corresponding to whether the target time $\tau^\star$ and/or the target set of recipients $S^\star$ must be disclosed before the Setup phase or not, and $i \in \{1, 2\}$, where when $i = 2$ the adversary is allowed to ask also master key queries for msk_τ of time τ while when $i = 1$ it is not. Note that the notion in [22] corresponds to $\mathsf{IND\text{-}aFS}_1\mathsf{\text{-}aSet\text{-}CCA}$ security. We note that a $\mathsf{IND\text{-}sFS}_i\mathsf{\text{-}ySet\text{-}CCA}$-secure scheme is also secure in the sense $\mathsf{IND\text{-}aFS}_i\mathsf{\text{-}ySet\text{-}CCA}$, albeit with the security degradation by factor T. For most applications, FS_1 security is sufficient. In this case, it is useful to consider the $\mathsf{MasUpdate}$ as a trivial algorithm as we let $\mathsf{msk}_\tau = \mathsf{msk}_0$ for all τ (and denote it by msk). Note that it is trivial to convert a scheme with FS_1 security to a new one achieving FS_2 security by letting msk_τ contains all user keys of time τ.

Conversion C [HICBE⇒FS-BE]. Given a HICBE scheme, we construct a FS-BE scheme using the "time tree" technique of [12], which was used to construct a forward-secure encryption from a binary tree encryption. Our conversion is essentially the same as that of [12] except that the user dimension is introduced.

For a forward-secure BE with T time periods, we image a complete balance binary tree of depth $L = \log_2(T+1) - 1$. Let each node be labeled with a string in $\{0,1\}^{\le L}$. We assign the root with the empty string. The left and right child of w is labeled $w0$ and $w1$ respectively. From now, to distinguish the abstract 'node' of a HICBE system from nodes in the binary tree, we refer to the former as h-node and the latter as usual. Following the notation in [12], we let w^τ to be the τ-th node in a pre-order traversal of the binary tree.[8] WLOG, we assume that $0, 1 \in \mathcal{I}$, the identity space. Hence, we can view a binary string of length $z \le L$ as an identity tuple of length z. Encryption in time τ for a set S of recipients uses the encryption function of the HICBE scheme to the multi-node (S, w^τ). At time τ the private key also contains, beside the private key of h-node (i, w^τ) of the HICBE scheme, all the keys of h-nodes (i, y) where y is a right sibling of the nodes on the path from the root to w^τ in the binary tree. When updating the key to time $\tau + 1$, we compute the private key of h-node $(i, w^{\tau+1})$ and erase the one of (i, w^τ). Since $w^{\tau+1}$ is a left child of w^τ or one of the nodes whose keys are stored as the additional keys at time τ, the derivation can be done, in particular, using at most one application of Derive. We denote this conversion as $\mathsf{C}(\cdot)$ and write its formal description and its security proof in [2].

Theorem 3. *Suppose that the scheme HICBE for L levels is $(t, q_{\mathsf{P}}, q_{\mathsf{D}}, \epsilon)$-$\mathsf{IND}$-$\mathsf{xID}$-$\mathsf{ySet}$-$\mathsf{CCA}$-secure (resp., $(t, q_{\mathsf{P}}, \epsilon)$-$\mathsf{IND}$-$\mathsf{xID}$-$\mathsf{ySet}$-$\mathsf{CPA}$-secure) for some $(\mathsf{x}, \mathsf{y}) \in \{(\mathsf{a}, \mathsf{a}), (\mathsf{a}, \mathsf{s}), (\mathsf{s}, \mathsf{a}), (\mathsf{s}, \mathsf{s})\}$. Then the scheme $\mathsf{C}(\mathsf{HICBE})$ for T time periods is $(t, q'_{\mathsf{P}}, q_{\mathsf{D}}, \epsilon)$-$\mathsf{IND}$-$\mathsf{xFS}_1$-$\mathsf{ySet}$-$\mathsf{CCA}$-secure (resp., $(t, q'_{\mathsf{P}}, \epsilon)$-$\mathsf{IND}$-$\mathsf{xFS}_1$-$\mathsf{ySet}$-$\mathsf{CPA}$-secure) for $q'_{\mathsf{P}} \le q_{\mathsf{P}}/L$, where $L = \log(T+1) - 1$.*

Resulting FS-BE Schemes. It is easy to see that in the resulting scheme, the private key size is expanded by the factor $O(\log T)$ while the other parameters are unchanged from the original HICBE scheme (instantiated for $\log(T+1)-1$ levels of identities). We have that the $\mathsf{C}(\mathsf{BasicHICBE1})$ scheme achieves ciphertext of size $O(\log T)$ and user private keys of size $O(\log^2 T)$ while the $\mathsf{C}(\mathsf{BasicHICBE2})$ scheme achieves ciphertexts of size $O(1)$ and user private keys of size $O(\log^2 T)$.

We also directly construct a more efficient but specific FS-BE scheme, denoted by $\mathsf{DirFSBE}$, which is not built via the generic conversion. It can be considered as a redundancy-free version of $\mathsf{C}(\mathsf{BasicHICBE1})$ which can reduce private key size to $O(\log T)$ without affecting other parameters. This can be seen as a reminiscent of the "Linear fs-HIBE" scheme in [5, full §C]. Its generalized scheme, denoted by $\mathsf{DirFSBE}(\mu)$, can be constructed as in §4.3. It trades off the public keys of size $O(n^\mu + n^{1-\mu} + \log T)$ with the ciphertexts of size $O(n^{1-\mu} + \log T)$.

Efficiency Comparisons. We draw comparisons among FS-BE schemes by wrapping up in Table 1. We name the three previous schemes intuitively from their approaches, where '$\times_{\mathsf{YFDL}}$' is the "cross-product" approach by Yao et al. [22], '$\perp_{\mathsf{BBG}}$' is the orthogonal integration approach by Boneh et al. [5, full §C], and the two operands indicate the underlying HIBEs, which include GS (the Gentry-Silverberg HIBE [17]), BB, and BBG. (See more details in [2]).

[8] The pre-order traversal is started from the root, $w^1 = \varepsilon$. From w it goes to $w0$ if w is not a leaf otherwise it goes to $v1$ if $v0$ is the largest string that is a prefix of w.

Table 1. Comparison among previous and our FS-BE schemes (upper and lower table resp.). $T = |\text{total time periods}|$. $n = |\text{all users}|$. $r = |\text{revoked users}|$. The time complexity is expressed in terms of number of operations where [e] is exponentiation, [p] is bilinear pairing, and [m] is group multiplication, while [o] indicates the time complexity for some other process. '$\Leftarrow$' means that it has the same value as the entry on its left.

Params$\downarrow$	$\mathsf{GS}_{(\mathsf{NNL})} \times_{\mathsf{YFDL}} \mathsf{GS}$ [22]	$\mathsf{BBG}_{(\mathsf{NNL})} \times_{\mathsf{YFDL}} \mathsf{BBG}$ [5, full §5.2]	$\mathsf{BBG}_{(\mathsf{NNL})} \perp_{\mathsf{BBG}} \mathsf{BB}$ [5, full §C]
Reg time	$O(\log^3 n \log T)$ [e]	$\Leftarrow$	$O((\log^2 n)(\log n + \log T))$ [e]
Enc time	$O(r \log n \log T)$ [e]	$\Leftarrow$	$O(r(\log n + \log T))$ [e]
Dec time	$O(\log n \log T)$ [p] $+ O(r)$ [o]	$\Leftarrow$	$O(\log T)$ [p] $+ O(r)$ [o]
Upd time	$O(\log^3 n)$ [e]	$\Leftarrow$	$O(\log^2 n \log T)$ [e]
|Pub key|	$O(\log n + \log T)$	$\Leftarrow$	$\Leftarrow$
|Pri key|	$O(\log^3 n \log T)$	$\Leftarrow$	$O((\log^2 n)(\log n + \log T))$
|Cipher|	$O(r \log n \log T)$	$O(r)$	$O(r \log T)$

Params$\downarrow$	$\mathsf{C}(\mathsf{BasicHICBE1})$	$\mathsf{DirFSBE}$	$\mathsf{C}(\mathsf{BasicHICBE2})$	$\mathsf{C}(\mathsf{GenHICBE}(0.5, 0.5))$
Reg time	$O(\log T)$ [e]	$\Leftarrow$	$\Leftarrow$	$O(\sqrt{\log T})$ [e]
Enc time	$O(n)$ [m] $+ O(\log T)$ [e]	$\Leftarrow$	$\Leftarrow$	$O(\sqrt{n})$ [m] $+ O(\sqrt{\log T})$ [e]
Dec time	$O(n)$ [m][9]$+ O(\log T)$ [p]	$\Leftarrow$	$O(n)$ [m][9]$+ O(1)$ [p]	$O(\sqrt{n})$ [m] $+ O(\sqrt{\log T})$ [p]
Upd time	$O(1)$ [e]	$\Leftarrow$	$\Leftarrow$	$\Leftarrow$
|Pub key|	$O(n + \log T)$	$\Leftarrow$	$\Leftarrow$	$O(\sqrt{n} + \sqrt{\log T})$
|Pri key|	$O(\log^2 T)$	$O(\log T)$	$O(\log^2 T)$	$O(\log^{1.5} T)$
|Cipher|	$O(\log T)$	$\Leftarrow$	$O(1)$	$O(\sqrt{n} + \sqrt{\log T})$

6 Public-key Broadcast Encryption with Keyword Search

6.1 Definitions and Relation to Anonymous ICBE

Model for BEKS. A public-key BE with keyword search (BEKS) consists of four algorithms. Via $(\mathsf{pk}, \{\mathsf{sk}_1, \ldots, \mathsf{sk}_n\}) \xleftarrow{R} \mathsf{Setup}(n)$ the setup algorithm produces a public key and n user keys; via $C \xleftarrow{R} \mathsf{BEKS}(\mathsf{pk}, S, \mathsf{w})$ a sender encrypts a keyword w to get a ciphertext (C, S) intended for recipients in $S \subseteq \{1, \ldots, n\}$; via $t_{i,\mathsf{w}} \xleftarrow{R} \mathsf{Td}(i, \mathsf{w}, \mathsf{sk}_i)$ the receiver i computes a trapdoor $(t_{i,\mathsf{w}}, i)$ for keyword w and provides it to the gateway (the server); via $b \leftarrow \mathsf{Test}(\mathsf{pk}, i, t_{i,\mathsf{w}}, C, S)$ for $i \in S$ the gateway can test whether C encrypts w where $b = 1$ means "positive" and $b = 0$ means "negative". Here if $i \notin S$ it always outputs '$\notin$'. We describe the right-keyword consistency (correctness), the computational consistency (in the sense of [1]), and the security notion, which we name IND-xKW-ySet-CPA, in the full paper [2]. The security captures the property that the adversary be unable to distinguish the encryption of chosen keyword with a random one.

Conversion K [ICBE$\Rightarrow$BEKS]. The conversion of [1] that compiles any anonymous IBE into a PEKS can be generalized to a broadcast version straightforwardly. More concretely, we construct BEKS from ICBE as follows. $\mathsf{Setup}_{\mathsf{BEKS}}(n)$ can be constructed from $\mathsf{Setup}_{\mathsf{ICBE}}$ and $\mathsf{PrivKeyGen}_{\mathsf{ICBE}}$ by relating the same public key pk and relating the private key $\mathsf{sk}_i = d_i$. The remaining algorithms work as

[9] This is due to the computation of $\prod_{j \in S, j \neq i} g_{n+1-j+i}$, which indeed can be pre-computed. This is useful when S is incrementally changed (cf. [7]).

follows: $t_{i,\mathsf{w}} \xleftarrow{R} \mathsf{Td}(i, \mathsf{w}, \mathsf{sk}_i) = \mathsf{Derive}_{\mathsf{ICBE}}(i, \mathsf{w}, d_i)$; $(C_1, C_2) \xleftarrow{R} \mathsf{BEKS}(\mathsf{pk}, S, \mathsf{w}) = \mathsf{Encrypt}_{\mathsf{ICBE}}(\mathsf{pk}, S, \mathsf{w})$; $\mathsf{Test}(\mathsf{pk}, i, t_{\mathsf{w}}, (C_1, C_2), S)$ outputs '$\notin$' if $i \notin S$, else outputs 1 if $\mathsf{Decrypt}_{\mathsf{ICBE}}(\mathsf{pk}, S, i, t_{\mathsf{w}}, C_1) = C_2$, else outputs 0. Denote this conversion by $\mathsf{K}(\cdot)$. Its correctness is immediate from that of ICBE. Indeed, $t_{i,\mathsf{w}}, C_1, C_2$ are related to $d_{i,\mathsf{w}}, \mathsf{hdr}, K$ in the ICBE scheme respectively. We remark that our conversion is a little bit different from (and simpler than) that of [1], in particular, since we have formalized the ICBE as a KEM.

Theorem 4. *(Informal) If the scheme* ICBE *is* ANO-xID-ySet-CPA$[\{1\}]$*-secure, then the BEKS scheme* K(ICBE) *is* IND-xKW-ySet-CPA*-secure. Further, if* ICBE *is semantically secure, then* K(ICBE) *is computationally consistent.*

6.2 Constructing Anonymous (H)ICBE

Attempts. As one may expect, the first attempt is to use our integration method to combine the BGW system with the anonymous HIBE, BW, by Boyen-Waters [11], which has a BB/BBG-like structure. Somewhat surprisingly and unfortunately, the resulting HICBE scheme is *not* ANO-sID-sSet-CPA-secure. Essentially, this is precisely due to the implicit orthogonality of BGW and BW. Such a property enables any user $i \notin S^\star$ to use the independent part of private keys corresponding to the BW portion to easily distinguish whether a ciphertext is intended for $(S^\star, \mathsf{ID}^\star)$ or $(S^\star, R)$ for random R, thus breaking anonymity. Dilemmatically, on the one hand, this orthogonality enables us to prove the confidentiality of the combined scheme; on the other hand, this very property gives an attack to the anonymity. We also remark that the approach $\mathsf{BB}_{(\mathsf{NNL})} \perp_{\mathsf{BBG}} \mathsf{BW}$ and $\mathsf{BBG}_{(\mathsf{NNL})} \perp_{\mathsf{BBG}} \mathsf{BW}$ (where notations are borrowed from the end of §5) also do *not* preserve the anonymity of BW due to a similar reason. See details in [2].

The Construction. From the above discussion, it is then natural to implement both the broadcast and identity dimensions from two non-orthogonal subsystems. Therefore, we construct our scheme in [2], denoted by AnonHICBE, from the YFDL (cross-product) approach instantiated to two copies of the BW hierarchies, or in our terminology, $\mathsf{BW}_{(\mathsf{NNL})} \times_{\mathsf{YFDL}} \mathsf{BW}$.[10] The resulting anonymous ICBE system achieves ciphertext of size $O(r \log n)$ and private key of size $O(\log^4 n)$ for the user level (level 0) and private key of size $O(\log^3 n)$ for level 1. These translate to the sizes of ciphertext, private key, and trapdoor in BEKS respectively.

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *Advances in Cryptology — CRYPTO 2005*, LNCS 3621, pp. 205-222. Springer, 2005.

[10] Probably this is the same approach as the one that Boyen and Waters used to construct an anonymous FS-HIBE, briefly mentioned in [11, full p.4], although it might be a different one (and we would not know since no detail was given there). One difference to mention is that the two hierarchies correspond to time/ID there, as opposed to broadcast/ID here.

2. N. Attrapadung, J. Furukawa, and H. Imai. Full version of this paper (with the same title). To be available at `http://eprint.iacr.org`.
3. N. Attrapadung and H. Imai. Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations. In *Advances in Cryptology — Asiacrypt 2005*, LNCS 3788 of LNCS, pp. 100-120. Springer, 2005.
4. D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Advances in Cryptology — Eurocrypt 2004*, LNCS 3027 of LNCS, pp. 223-238. Springer, 2004.
5. D. Boneh, X. Boyen, and E-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology — Eurocrypt 2005*, LNCS 3494, pp. 440-456. Springer, 2005. Full version available at `http://eprint.iacr.org/2005/015`.
6. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology — Eurocrypt 2004*, LNCS 3027, pp. 506-522. Springer, 2004.
7. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology — Crypto 2005*, LNCS 3621, pp. 258-275. Springer, 2005.
8. D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology — Crypto 2001*, LNCS 2139, pp. 213-229. Springer, 2001.
9. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *RSA-CT 2005*, LNCS 3376, pp. 87-103. 2005.
10. X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *Proc. ACM-CCS 2005*, pp. 320-329, ACM Press, 2005.
11. X. Boyen, B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Advances in Cryptology — Crypto 2006*, LNCS 4117, pp. 290-307. Springer, 2006. Full version available at `http://eprint.iacr.org/2006/085`.
12. R. Canetti, S. Halevi, J. Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology — Eurocrypt 2003*, LNCS 2656, pp. 255-271. 2003.
13. R. Canetti, S. Halevi, J. Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology — Eurocrypt 2004*, LNCS 3027, pp. 207-222.
14. J. H. Cheon. Security analysis of the strong Diffie-Hellman problem. In *Advances in Cryptology — Eurocrypt 2006*, LNCS 4004, pp. 1-11, Springer, 2006.
15. Y. Dodis and N. Fazio. Public-key broadcast encryption for stateless receivers. In *ACM Digital Rights Management 2002*, LNCS 2696, pp. 61-80. Springer, 2002.
16. A. Fiat and M. Naor. Broadcast encryption. In *Advances in Cryptology — Crypto 1993*, LNCS 773, pp. 480-491. Springer, 1993.
17. C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In *Advances in Cryptology — Asiacrypt 2002*, LNCS 2501, pp. 548-566. Springer, 2002.
18. M. T. Goodrich, J. Z. Sun, and R. Tamassia. Efficient tree-based revocation in groups of low-state devices. In *Advances in Cryptology — Crypto 2004*, LNCS 3152, pp. 511-527. Springer, 2004.
19. D. Halevy and A. Shamir. The LSD broadcast encryption scheme. In *Advances in Cryptology — Crypto 2002*, LNCS 2442, pp. 47-60. Springer, 2002.
20. D. Naor, M. Naor, J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Advances in Cryptology — Crypto 2001*, LNCS 2139, pp. 41-62. 2001.
21. B. Waters. Efficient identity-Based encryption without random oracles. In *Advances in Cryptology — Eurocrypt 2005*, LNCS 3494, pp. 114-127. Springer, 2005.
22. D. Yao, N. Fazio, Y. Dodis, and A. Lysyanskaya. ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In *Proc. ACM-CCS 2004*, pp. 354-363, ACM, 2004.