Fully Secure Anonymous HIBE and Secret-Key Anonymous IBE with Short Ciphertexts

Angelo De Caro[*] Vincenzo Iovino [†] Giuseppe Persiano [‡]

Thursday 28th October, 2010

Abstract

Lewko and Waters [Eurocrypt 2010] presented a fully secure HIBE with short ciphertexts. In this paper we show how to modify their construction to achieve anonymity. We prove the security of our scheme under static (and generically secure) assumptions formulated in composite order bilinear groups.

In addition, we present a fully secure Anonymous IBE in the secret-key setting. Secret-Key Anonymous IBE was implied by the work of [Shen-Shi-Waters - TCC 2009] which can be shown secure in the selective-id model. No previous fully secure construction of secret-key Anonymous IBE is known.

[*]Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84084 Fisciano (SA), Italy.

[†]Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84084 Fisciano (SA), Italy.. Work done while visiting the Department of Computer Science of The Johns Hopkins University.

[‡]Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84084 Fisciano (SA), Italy.

Contents

1 Introduction

Identity-Based Encryption (IBE) was introduced by [13] to simplify the public-key infrastructure. An IBE is a public-key encryption scheme in which the public-key can be set to any string interpreted as one's identity. A central authority that holds the master secret key can produce a secret key corresponding to a given identity. Anyone can then encrypt messages using the identity, and only the owner of the corresponding secret key can decrypt the messages. First realizations of IBE are due to [2] which makes use of bilinear groups and to [8] which uses quadratic residues. Later, [9] introduced the more general concept of Hierarchical Identity-Based Encryption (HIBE) issuing a partial solution to it. An HIBE system is an IBE that allows delegation of the keys in a hierarchical structure. To the top of the structure there is the central authority that holds the master secret key, then several sub-authorities (or individual users) that hold delegated keys which can be used to decrypt only the messages addressed to the organization which the sub-authority belongs.

In this paper we are interested in *Anonymous* HIBE that are a special type of HIBE with the property that ciphertexts hide the identity for which they were encrypted. Interest in Anonymous IBE and HIBE was spurred by the observation that it can be used to build Public-Key Encryption with Keyword Search [1]. As noticed by [4], the first construction of Anonymous IBE was implicit in [2] whose security relied on the random oracle assumption. Boneh and Waters [5] constructed Anonymous HIBE in the selective-id model. Recently, Lewko and Waters [11] used the Dual System Encryption methodology introduced by [16] to construct the first fully secure HIBE system with short ciphertexts. The construction given by [11] seems inherently non-anonymous. Another construction of Anonymous HIBE was given by [12] but their security proof is in the selective-ID model.

We show that a slight modification of the HIBE of [11] gives the *first* fully secure Anonymous HIBE. Our construction has, like the non-anonymous one of [11], short ciphertexts; that is, a ciphertext consists of a constant (that is independent of the depth of the hierarchy) number of elements from the underlying bilinear group. The full security of our construction is based on static (that is, independent from the running time of the adversary and the size of hierarchy) and generically secure assumptions.

Recently, a fully-secure hierarchical predicate encryption system has been given by [10]. Anonymous HIBE can be obtained as special case of the construction of [10] and, even though the construction of [10] is based on prime order gropus, the ciphertexts of the resulting Anonymous HIBE consist of $O(\ell^2)$ group elements and keys have $O(\ell^3)$ group elements. In [7] the authors constructed an Anonymous HIBE scheme based on hard lattice problems; in this construction the size of a ciphertext depends on the depth of the hierarchy.

We also study *Secret-Key* Anonymous IBE and show that if our public key construction is used in the secret key setting (that is, the public key is kept secret) then the scheme enjoys the additional property of key secrecy; that is, decryption keys for different identities are indistinguishable. We stress that key secrecy cannot be obtained in the public key setting as an adversary can test if a secret key Sk corresponds to identity ID by creating a cipertext Ct for ID (by using the *public key*) and then trying to decrypt Ct by using Sk. We mention that the Secret-Key Predicate Encryption Scheme of [14] has Secret-Key Anonymous IBE as a special case but its security is in the selective-id model. To the best of our knowledge, the concept of Secret-Key Hierarchical IBE has not been defined before and we defer its study to future work.

Organization of the work. In Section 2, we present a brief introduction of bilinear groups and state the complexity assumptions used to prove the security of our schemes. In Section 3 we present definitions and our construction for Public-key Anonymous HIBE. Finally, in Section 4, we present definitions and our construction for Secret-key Anonymous IBE. Due to lack of space the proof of security of our assumptions in the generic group model is omitted and can be found in the extended version [6].

2 Composite Order Bilinear Groups and Complexity Assumptions

Composite order bilinear groups were first used in cryptographic construction in [3]. We use groups of order product of four primes and a generator $\mathcal{G}$ which takes as input security parameter λ and outputs a description $\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ where p_1, p_2, p_3, p_4 are distinct primes of $\Theta(\lambda)$ bits, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order N, and $\mathbf{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a map with the following properties:

1. (Bilinearity) $\forall\ g, h \in \mathbb{G}, a, b \in \mathbb{Z}_N, \mathbf{e}(g^a, h^b) = \mathbf{e}(g, h)^{ab}$.
2. (Non-degeneracy) $\exists\ g \in \mathbb{G}$ such that $\mathbf{e}(g, g)$ has order N in $\mathbb{G}_T$.

We further require that the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well the bilinear map $\mathbf{e}$ are computable in deterministic polynomial time with respect to λ. Also, we assume that the group descriptions of $\mathbb{G}$ and $\mathbb{G}_T$ include generators of the respective cyclic groups. Furthermore, for $a, b, c \in \{1, p_1, p_2, p_3, p_4\}$ we denote by $\mathbb{G}_{abc}$ the subgroup of order abc. From the fact that the group is cyclic it is simple to verify that if g and h are group elements of different order (and thus belonging to different subgroups), then $\mathbf{e}(g, h) = 1$. This is called the *orthogonality property* and is a crucial tool in our constructions. We now give our complexity assumptions.

2.1 Assumption 1

For a generator $\mathcal{G}$ returning bilinear settings of order N product of four primes, we define the following distribution. First pick a random bilinear setting $\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ by running $\mathcal{G}(1^\lambda)$ and then pick

$$g_1, A_1 \leftarrow \mathbb{G}_{p_1}, A_2, B_2 \leftarrow \mathbb{G}_{p_2}, g_3, B_3 \leftarrow \mathbb{G}_{p_3}, g_4 \leftarrow \mathbb{G}_{p_4}, T_1 \leftarrow \mathbb{G}_{p_1p_2p_3}, T_2 \leftarrow \mathbb{G}_{p_1p_3}$$

and set $D = (\mathcal{I}, g_1, g_3, g_4, A_1A_2, B_2B_3)$. We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be:

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{A1}}(\lambda) = |\text{Prob}[\mathcal{A}(D, T_1) = 1] - \text{Prob}[\mathcal{A}(D, T_2) = 1]|.$$

Assumption 1 *We say that Assumption* 1 *holds for generator $\mathcal{G}$ if for all probabilistic polynomial-time algorithms $\mathcal{A}$ $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{A1}}(\lambda)$ is a negligible function of λ.*

2.2 Assumption 2

For a generator $\mathcal{G}$ returning bilinear settings of order N product of four primes, we define the following distribution. First pick a random bilinear setting $\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ by running $\mathcal{G}(1^\lambda)$ and then pick

$$\alpha, s, r \leftarrow \mathbb{Z}_N,\ g_1 \leftarrow \mathbb{G}_{p_1},\ g_2, A_2, B_2 \leftarrow \mathbb{G}_{p_2},\ g_3 \leftarrow \mathbb{G}_{p_3},\ g_4 \leftarrow \mathbb{G}_{p_4},\ T_2 \leftarrow \mathbb{G}_T$$

and set $T_1 = \mathbf{e}(g_1, g_1)^{\alpha s}$ and $D = (\mathcal{I}, g_1, g_2, g_3, g_4, g_1^{\alpha} A_2, g_1^s B_2, g_2^r, A_2^r)$. We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be:

$$\mathsf{Adv}_{\mathsf{A2}}^{\mathcal{A}}(\lambda) = |\text{Prob}[\mathcal{A}(D, T_1) = 1] - \text{Prob}[\mathcal{A}(D, T_2) = 1]|.$$

Assumption 2 *We say that Assumption* 2 *holds for generator* $\mathcal{G}$ *if for all probabilistic polynomial time algorithm* $\mathcal{A}$ $\mathsf{Adv}_{\mathsf{A2}}^{\mathcal{A}}(\lambda)$ *is a negligible function of* λ.

2.3 Assumption 3

For a generator $\mathcal{G}$ returning bilinear settings of order N product of four primes, we define the following distribution. First pick a random bilinear setting $\mathcal{I} = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ by running $\mathcal{G}(1^\lambda)$ and then pick

$$\hat{r}, s \leftarrow \mathbb{Z}_N,\ g_1, U, A_1 \leftarrow \mathbb{G}_{p_1},\ g_2, A_2, B_2, D_2, F_2 \leftarrow \mathbb{G}_{p_2},\ g_3 \leftarrow \mathbb{G}_{p_3},$$
$$g_4, A_4, B_4, D_4 \leftarrow \mathbb{G}_{p_4},\ A_{24}, B_{24}, D_{24} \leftarrow \mathbb{G}_{p_2 p_4},\ T_2 \leftarrow \mathbb{G}_{p_1 p_2 p_4}$$

and set $T_1 = A_1^s D_{24}$ and $D = (\mathcal{I}, g_1, g_2, g_3, g_4, U, U^s A_{24}, U^{\hat{r}}, A_1 A_4, A_1^{\hat{r}} A_2, g_1^{\hat{r}} B_2,\ g_1^s B_{24})$. We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be:

$$\mathsf{Adv}_{\mathsf{A3}}^{\mathcal{A}}(\lambda) = |\text{Prob}[\mathcal{A}(D, T_1) = 1] - \text{Prob}[\mathcal{A}(D, T_2) = 1]|.$$

Assumption 3 *We say that Assumption* 3 *holds for generator* $\mathcal{G}$ *if for all probabilistic polynomial time algorithm* $\mathcal{A}$ $\mathsf{Adv}_{\mathsf{A3}}^{\mathcal{A}}(\lambda)$ *is a negligible function of* λ.

3 Public-Key Anonymous HIBE

3.1 Hierarchical Identity Based Encryption

A Hierarchical Identity Based Encryption scheme (henceforth abbreviated in HIBE) over an alphabet Σ is a tuple of five efficient and probabilistic algorithms: (Setup, Encrypt, KeyGen, Decrypt, Delegate).

Setup$(1^\lambda, 1^\ell)$: takes as input security parameter λ and maximum depth of an identity vector ℓ and outputs public parameters Pk and master secret key Msk.

KeyGen(Msk, ID $= (\mathsf{ID}_1, \ldots, \mathsf{ID}_j)$): takes as input master secret key Msk, identity vector $ID \in \Sigma^j$ with $j \le \ell$ and outputs a private key $\mathsf{Sk}_{\mathsf{ID}}$.

Delegate(Pk, ID, $\mathsf{Sk}_{\mathsf{ID}}$, ID_{j+1}): takes as input public parameters Pk, secret key for identity vector ID $= (\mathsf{ID}_1, \ldots, \mathsf{ID}_j)$ of depth $j < \ell$, $\mathsf{ID}_{j+1} \in \Sigma$ and outputs a secret key for the depth $j+1$ identity vector $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j, \mathsf{ID}_{j+1})$.

Encrypt(Pk, M, ID): takes as input public parameters Pk, message M and identity vector ID and outputs a ciphertext Ct.

Decrypt(Pk, Ct, Sk): takes as input public parameters Pk, ciphertext Ct and secret key Sk and outputs the message M. We make the following obvious consistency requirement. Suppose ciphertext Ct is obtained by running the Encrypt algorithm on public parameters Pk, message M and identity ID and that Sk is a secret key for identity ID obtained through a sequence of KeyGen and Delegate calls using the same public parameters Pk. Then Decrypt, on input Pk, Ct and Sk, returns M except with negligible probability.

3.2 Security definition

We give complete form of the security definition following [15]. Our security definition captures semantic security and ciphertext anonymity by means of the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Setup. The challenger $\mathcal{C}$ runs the Setup algorithm to generate public parameters Pk which it gives to the adversary $\mathcal{A}$. We let S denote the set of private keys that the challenger has created but not yet given to the adversary. At this point, $S = \emptyset$.

Phase 1. $\mathcal{A}$ makes Create, Delegate, and Reveal key queries. To make a Create query, $\mathcal{A}$ specifies an identity vector ID of depth j. In response, $\mathcal{C}$ creates a key for this vector by calling the key generation algorithm, and places this key in the set S. $\mathcal{C}$ only gives $\mathcal{A}$ a reference to this key, not the key itself. To make a Delegate query, $\mathcal{A}$ specifies a key $\mathsf{Sk}_{\mathsf{ID}}$ in the set S and $\mathsf{ID}_{j+1} \in \Sigma$. In response, $\mathcal{C}$ appends ID_{j+1} to ID and makes a key for this new identity by running the delegation algorithm on $\mathsf{ID}, \mathsf{Sk}_{\mathsf{ID}}$ and ID_{j+1}. $\mathcal{C}$ adds this key to the set S and again gives $\mathcal{A}$ only a reference to it, not the actual key. To make a Reveal query, $\mathcal{A}$ specifies an element of the set S. $\mathcal{C}$ gives this key to $\mathcal{A}$ and removes it from the set S. We note that $\mathcal{A}$ needs no longer make any delegation queries for this key because it can run delegation algorithm on the revealed key for itself.

Challenge. $\mathcal{A}$ gives two pairs of message and identity $(M_0, \mathsf{ID}_0^\star)$ and $(M_1, \mathsf{ID}_1^\star)$ to $\mathcal{C}$. We require that no revealed identity in Phase 1 is a prefix of either $\mathsf{ID}_0^\star$ or $\mathsf{ID}_1^\star$. $\mathcal{C}$ chooses random $\beta \in \{0, 1\}$, encrypts M_β under $\mathsf{ID}_\beta^\star$ and sends the resulting ciphertext to $\mathcal{A}$.

Phase 2. This is the same as Phase 1 with the added restriction that any revealed identity vector must not be a prefix of either $\mathsf{ID}_0^\star$ or $\mathsf{ID}_1^\star$.

Guess. $\mathcal{A}$ must output a guess β' for β. The advantage of $\mathcal{A}$ is defined to be $\text{Prob}[\beta' = \beta] - \frac{1}{2}$.

Definition 3.1 *An Anonymous Hierarchical Identity Based Encryption scheme is secure if all polynomial time adversaries achieve at most a negligible (in λ) advantage in the previous security game.*

3.3 Our construction

In this section we describe our construction for an Anonymous HIBE scheme.

$\mathsf{Setup}(1^\lambda, 1^\ell)$: The setup algorithm chooses random description $\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ and random $Y_1, X_1, u_1, \ldots, u_\ell \in \mathbb{G}_{p_1}, Y_3 \in \mathbb{G}_{p_3}, X_4, Y_4 \in \mathbb{G}_{p_4}$ and $\alpha \in \mathbb{Z}_N$. The public parameters are published as:

$$\mathsf{Pk} = (N, Y_1, Y_3, Y_4, t = X_1X_4, u_1, \ldots, u_\ell, \Omega = \mathbf{e}(Y_1, Y_1)^\alpha).$$

The master secret key is $\mathsf{Msk} = (X_1, \alpha)$.

$\mathsf{KeyGen}(\mathsf{Msk}, \mathsf{ID} = (\mathsf{ID}_1, \ldots, \mathsf{ID}_j))$: The key generation algorithm chooses random $r_1, r_2 \in \mathbb{Z}_N$ and, for $i \in \{1,2\}$, random $R_{i,1}, R_{i,2}, R_{i,j+1}, \ldots, R_{i,\ell} \in \mathbb{G}_{p_3}$. The secret key $\mathsf{Sk}_{\mathsf{ID}} = (K_{i,1}, K_{i,2}, E_{i,j+1}, \ldots, E_{i,\ell})$ is computed as

$$K_{1,1} = Y_1^{r_1} R_{1,1}, \quad K_{1,2} = Y_1^{\alpha} \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1 \right)^{r_1} R_{1,2}$$

$$E_{1,j+1} = u_{j+1}^{r_1} R_{1,j+1}, \; \ldots, \; E_{1,\ell} = u_\ell^{r_1} R_{1,\ell},$$

$$K_{2,1} = Y_1^{r_2} R_{2,1}, \quad K_{2,2} = \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1 \right)^{r_2} R_{2,2}$$

$$E_{2,j+1} = u_{j+1}^{r_2} R_{2,j+1}, \; \ldots, \; E_{1,\ell} = u_\ell^{r_2} R_{2,\ell}.$$

Notice that, $\mathsf{Sk}_{\mathsf{ID}}$ is composed by two sub-keys. The first sub-key, $(K_{1,1}, K_{1,2}, E_{1,j+1}, \ldots, E_{1,\ell})$, is used by the decryption algorithm to compute the blinding factor, the second, $(K_{2,1}, K_{2,2}, E_{2,j+1}, \ldots, E_{2,\ell})$, is used by the delegation algorithm and can be used also to verify that the identity vector of a given ciphertext matches the identity vector of the key.

$\mathsf{Delegate}(\mathsf{Pk}, \mathsf{ID}, \mathsf{Sk}_{\mathsf{ID}}, \mathsf{ID}_{j+1})$: Given a key $\mathsf{Sk}_{\mathsf{ID}} = (K'_{i,1}, K'_{i,2}, E'_{i,j+1}, \ldots, E'_{i,\ell})$ for $\mathsf{ID} = (\mathsf{ID}_1, \ldots, \mathsf{ID}_j)$, the delegation algorithm creates a key for $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j, \mathsf{ID}_{j+1})$ as follows. It chooses random $\tilde{r_1}, \tilde{r_2} \in \mathbb{Z}_N$ and, for $i \in \{1,2\}$, random $R_{i,1}, R_{i,2}, R_{i,j+2}, \ldots, R_{i,\ell} \in \mathbb{G}_{p_3}$. The secret key $(K_{i,1}, K_{i,2}, E_{i,j+2}, \ldots, E_{i,\ell})$ is computed as

$$K_{1,1} = K'_{1,1}(K'_{2,1})^{\tilde{r_1}} R_{1,1}, K_{1,2} = K'_{1,2}(K'_{2,2})^{\tilde{r_1}} \left(E'_{1,j+1}\right)^{\mathsf{ID}_{j+1}} \left(E'_{2,j+1}\right)^{\tilde{r_1}\mathsf{ID}_{j+1}} R_{1,2},$$

$$E_{1,j+2} = E'_{1,j+2} \cdot (E'_{2,j+2})^{\tilde{r_1}} R_{1,j+2}, \ldots, E_{1,\ell} = E'_{1,\ell} \cdot (E'_{2,\ell})^{\tilde{r_1}} R_{1,\ell}.$$

and

$$K_{2,1} = (K'_{2,1})^{\tilde{r_2}} R_{2,1}, \quad K_{2,2} = (K'_{2,2})^{\tilde{r_2}} \cdot (E'_{2,j+1})^{\tilde{r_2}\mathsf{ID}_{j+1}} R_{2,2},$$

$$E_{2,j+2} = (E'_{2,j+2})^{\tilde{r_2}} R_{2,j+2}, \ldots, E_{2,\ell} = (E'_{2,\ell})^{\tilde{r_2}} R_{2,\ell}.$$

We observe that the new key has the same distributions as the key computed by the KeyGen algorithm on $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j, \mathsf{ID}_{j+1})$ with randomness $r_1 = r'_1 + (r'_2 \cdot \tilde{r_1})$ and $r_2 = r'_2 \cdot \tilde{r_2}$.

$\mathsf{Encrypt}(\mathsf{Pk}, M, \mathsf{ID} = (\mathsf{ID}_1, \ldots, \mathsf{ID}_j))$: The encryption algorithm chooses random $s \in \mathbb{Z}_N$ and random $Z, Z' \in \mathbb{G}_{p_4}$. The ciphertext (C_0, C_1, C_2) for the message $M \in \mathbb{G}_T$ is computed as

$$C_0 = M \cdot \mathbf{e}(Y_1, Y_1)^{\alpha s}, \quad C_1 = \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} t \right)^s Z, \quad C_2 = Y_1^s Z'.$$

$\mathsf{Decrypt}(\mathsf{Pk}, \mathsf{Ct}, \mathsf{Sk})$: The decryption algorithm assumes that the key and ciphertext both correspond to the same identity $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j)$. If the key identity is a prefix of this instead, then the decryption algorithm starts by running the key delegation algorithm to create a key with identity matching the ciphertext identity exactly. The decryption algorithm then computes the blinding factor as:

$$\frac{\mathbf{e}(K_{1,2}, C_2)}{\mathbf{e}(K_{1,1}, C_1)} = \frac{\mathbf{e}(Y_1, Y_1)^{\alpha s} \mathbf{e}\left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1, Y_1 \right)^{r_1 s}}{\mathbf{e}\left(Y_1, u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1 \right)^{r_1 s}} = \mathbf{e}(Y_1, Y_1)^{\alpha s}.$$

By comparing our construction with the one of [11], we notice that component t of the public key and components C_1 and C_2 of the ciphertext have a $\mathbb{G}_{p_4}$ part. This addition makes the system anonymous. Indeed, if we remove from our construction the $\mathbb{G}_{p_4}$ parts of t and C_1 and C_2 (and thus obtain the scheme of [11]) then it is possible to test if ciphertext (C_0, C_1, C_2) is relative to identity $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j)$ for public key $(N, Y_1, Y_3, Y_4, t, u_1, \ldots, u_\ell, \Omega)$ by testing $\mathbf{e}(C_2, t \cdot (u_1^{\mathsf{ID}_1} \cdots u_\ell^{\mathsf{ID}_\ell}))$ and $\mathbf{e}(C_1, Y_1)$ for equality.

3.4 Security

Following Lewko and Waters [11], we define two additional structures: *semi-functional ciphertexts* and *semi-functional keys*. These will not be used in the real scheme, but we need them in our proofs.

Semi-functional Ciphertext. We let g_2 denote a generator of $\mathbb{G}_{p_2}$. A semi-functional ciphertext is created as follows: first, we use the encryption algorithm to form a normal ciphertext (C_0', C_1', C_2'). We choose random exponents $x, z_c \in \mathbb{Z}_N$. We set:

$$C_0 = C_0', \quad C_1 = C_1' g_2^{x z_c}, \quad C_2 = C_2' g_2^{x}.$$

Semi-functional Keys. To create a semi-functional key, we first create a normal key $(K_{i,1}', K_{i,2}', E_{i,j+1}', \ldots, E_{i,\ell}')$ using the key generation algorithm. We choose random exponents $z, \gamma, z_k \in \mathbb{Z}_N$ and, for $i \in \{1, 2\}$, random exponents $z_{i,j+1}, \ldots, z_{i,\ell} \in \mathbb{Z}_N$. We set:

$$K_{1,1} = K_{1,1}' \cdot g_2^{\gamma}, K_{1,2} = K_{1,2}' \cdot g_2^{\gamma z_k}, (E_{1,i} = E_{1,i}' \cdot g_2^{\gamma z_{1,i}})_{i=j+1}^{\ell},$$

and

$$K_{2,1} = K_{2,1}' \cdot g_2^{z\gamma}, K_{2,2} = K_{2,2}' \cdot g_2^{z\gamma z_k}, (E_{2,i} = E_{2,i}' \cdot g_2^{z\gamma z_{2,i}})_{i=j+1}^{\ell}.$$

We note that when the first sub-key of a semi-functional key is used to decrypt a semi-functional ciphertext, the decryption algorithm will compute the blinding factor multiplied by the additional term $\mathbf{e}(g_2, g_2)^{x\gamma(z_k - z_c)}$. If $z_c = z_k$, decryption will still work. In this case, we say that the key is nominally semi-functional. If the second sub-key is used to test the identity vector of the ciphertext, then the decryption algorithm computes $\mathbf{e}(g_2, g_2)^{xz\gamma(z_k - z_c)}$ and if $z_c = z_k$, the test will still work.

To prove security of our Anonymous HIBE scheme, we rely on the static Assumptions 1, 2 and 3. For a probabilistic polynomial-time adversary $\mathcal{A}$ which makes q key queries, our proof of security will consist of the following sequence of $q + 5$ games between $\mathcal{A}$ and a challenger $\mathcal{C}$.

$\mathsf{Game}_{\mathsf{Real}}$: is the real Anonymous HIBE security game.

$\mathsf{Game}_{\mathsf{Real}'}$: is the same as the real game except that all key queries will be answered by fresh calls to the key generation algorithm, ($\mathcal{C}$ will not be asked to delegate keys in a particular way).

$\mathsf{Game}_{\mathsf{Restricted}}$: is the same as $\mathsf{Game}_{\mathsf{Real}'}$ except that $\mathcal{A}$ cannot ask for keys for identities which are prefixes of one of the challenge identities modulo p_2. We will retain this restriction in all subsequent games.

Game_k: for k from 0 to q, we define Game_k like $\mathsf{Game}_{\mathsf{Restricted}}$ except that the ciphertext given to $\mathcal{A}$ is semi-functional and the first k keys are semi-functional. The rest of the keys are normal.

$\mathsf{Game}_{\mathsf{Final}_0}$: is the same as Game_q, except that the challenge ciphertext is a semi-functional encryption with C_0 random in $\mathbb{G}_T$ (thus the ciphertext is independent from the messages provided by $\mathcal{A}$).

$\mathsf{Game}_{\mathsf{Final}_1}$: is the same as $\mathsf{Game}_{\mathsf{Final}_0}$, except that the challenge ciphertext is a semi-functional encryption with C_1 random in $\mathbb{G}_{p_1p_2p_4}$ (thus the ciphertext is independent from the identity vectors provided by $\mathcal{A}$). It is clear that in this last game, no adversary can have advantage greater than 0.

We will show these games are indistinguishable in the following lemmata.

3.4.1 Indistinguishability of $\mathsf{Game}_{\mathsf{Real}}$ and $\mathsf{Game}_{\mathsf{Real}'}$

Lemma 3.2 *For any algorithm $\mathcal{A}$, $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Real}}} = \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Real}'}}$.*

Proof. We note that the keys are identically distributed whether they are produced by the key delegation algorithm from a previous key or from a fresh call to the key generation algorithm. Thus, in the attacker's view, there is no difference between these games. $\square$

3.4.2 Indistinguishability of $\mathsf{Game}_{\mathsf{Real}'}$ and $\mathsf{Game}_{\mathsf{Restricted}}$

Lemma 3.3 *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Real}'}} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Restricted}}} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage $\geq \frac{\epsilon}{3}$ in breaking Assumption 1.*

Proof. Suppose that $\mathcal{A}$ has probability ϵ of producing an identity vector $\mathsf{ID} = (\mathsf{ID}_1, \ldots, \mathsf{ID}_k)$, that is a prefix of one of the challenge identities $\mathsf{ID}^\star = (\mathsf{ID}^\star_1, \ldots, \mathsf{ID}^\star_j)$ modulo p_2. That is, there exists i and $j \in \{0,1\}$ such that that $\mathsf{ID}_i \neq \mathsf{ID}^\star_{j,i}$ modulo N and that p_2 divides $\mathsf{ID}_i - \mathsf{ID}^\star_{j,i}$ and thus $a = \gcd(\mathsf{ID}_i - \mathsf{ID}^\star_{j,i}, N)$ is a nontrivial factor of N. We notice that p_2 divides a and set $b = \frac{N}{a}$. The following three cases are exhaustive and at least one occurs with probability at least $\epsilon/3$.

1. $\mathrm{ord}(Y_1) \mid b$.
2. $\mathrm{ord}(Y_1) \nmid b$ and $\mathrm{ord}(Y_4) \mid b$.
3. $\mathrm{ord}(Y_1) \nmid b$, $\mathrm{ord}(Y_4) \nmid b$ and $\mathrm{ord}(Y_3) \mid b$.

Suppose case 1 has probability at least $\epsilon/3$. We describe algorithm $\mathcal{B}$ that breaks Assumption 1. $\mathcal{B}$ receives $(\mathcal{I}, g_1, g_3, g_4, A_1A_2, B_2B_3)$ and T and constructs Pk by running the Setup algorithm with the only exception that $\mathcal{B}$ sets $Y_1 = g_1, Y_3 = g_3$, and $Y_4 = g_4$. Notice that $\mathcal{B}$ has the master secret key Msk associated with Pk. Then $\mathcal{B}$ runs $\mathcal{A}$ on input Pk and uses knowledge of Msk to answer $\mathcal{A}$'s queries. At the end of the game, for all IDs for which $\mathcal{A}$ has asked for the key and for $\mathsf{ID}^\star \in \{\mathsf{ID}^\star_0, \mathsf{ID}^\star_1\}$, $\mathcal{B}$ computes $a = \gcd(\mathsf{ID}_i - \mathsf{ID}^\star_i, N)$. Then, if $\mathbf{e}\left((A_1A_2)^a, B_2B_3\right)$ is the identity element of $\mathbb{G}_T$ then $\mathcal{B}$ tests if $\mathbf{e}(T^b, A_1A_2)$ is the identity element of $\mathbb{G}_T$. If this second test is successful, then $\mathcal{B}$ declares $T \in \mathbb{G}_{p_1p_3}$. If it is not, $\mathcal{B}$ declares $T \in \mathbb{G}_{p_1p_2p_3}$. It is easy to see that if p_2 divides a and $p_1 = \mathrm{ord}(Y_1)$ divides b, then $\mathcal{B}$'s output is correct.

The other two cases are similar. Specifically, in case 2, $\mathcal{B}$ breaks Assumption 1 in the same way except that Pk is constructed by setting $Y_1 = g_4, Y_3 = g_3$, and $Y_4 = g_1$ (this has the effect of exchanging the roles of p_1 and p_4). Instead in case 3, $\mathcal{B}$ constructs Pk by setting $Y_1 = g_3, Y_3 = g_1$, and $Y_4 = g_4$ (this has the effect of exchanging the roles of p_1 and p_3). $\square$

3.4.3 Indistinguishability of $\mathsf{Game}_{\mathsf{Restricted}}$ and Game_0

Lemma 3.4 *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Restricted}}} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_0} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 1.

Proof. $\mathcal{B}$ receives $(\mathcal{I}, g_1, g_3, g_4, A_1A_2, B_2B_3)$ and T and simulates $\mathsf{Game}_{\mathsf{Restricted}}$ or Game_0 with $\mathcal{A}$ depending on whether $T \in \mathbb{G}_{p_1p_3}$ or $T \in \mathbb{G}_{p_1p_2p_3}$.

$\mathcal{B}$ sets the public parameters as follows. $\mathcal{B}$ chooses random exponents $\alpha, a_1, \ldots, a_\ell, b, c \in \mathbb{Z}_N$ and sets $Y_1 = g_1, Y_3 = g_4, Y_4 = g_3$, $X_4 = Y_4^c, X_1 = Y_1^b$ and $u_i = Y_1^{a_i}$ for $i \in [\ell]$. $\mathcal{B}$ sends $\mathsf{Pk} = (N, Y_1, Y_3, Y_4, t = X_1X_4, u_1, \ldots, u_\ell, \Omega = \mathbf{e}(Y_1, Y_1)^\alpha)$ to $\mathcal{A}$. Notice that $\mathcal{B}$ knows the master secret key $\mathsf{Msk} = (X_1, \alpha)$ associated with Pk and thus can answer all $\mathcal{A}$'s queries.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two pairs, $(M_0, \mathsf{ID}_0^\star = (\mathsf{ID}_{0,1}^\star, \ldots, \mathsf{ID}_{0,j}^\star))$ and $(M_1, \mathsf{ID}_1^\star = (\mathsf{ID}_{1,1}^\star, \ldots, \mathsf{ID}_{1,j}^\star))$. $\mathcal{B}$ chooses random $\beta \in \{0,1\}$ and computes the challenge ciphertext as follows:

$$C_0 = M_\beta \cdot \mathbf{e}(T, Y_1)^\alpha, \quad C_1 = T^{a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b}, \quad C_2 = T.$$

We complete the proof with the following two observations. If $T \in \mathbb{G}_{p_1p_3}$, then T can be written as $Y_1^{s_1}Y_4^{s_3}$. In this case (C_0, C_1, C_2) is a normal ciphertext with randomness $s = s_1, Z = Y_4^{s_3a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b}$ and $Z' = Y_4^{s_3}$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then T can be written as $Y_1^{s_1}g_2^{s_2}Y_4^{s_3}$ and this case (C_0, C_1, C_2) is a semi-functional ciphertext with randomness $s = s_1, Z = Y_4^{s_3a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b}$, $Z' = Y_4^{s_3}$, $\gamma = s_2$ and $z_c = a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b$. $\square$

3.4.4 Indistinguishability of Game_{k-1} and Game_k

Lemma 3.5 *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{k-1}} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_k} = \epsilon$. Then, there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 1.

Proof. $\mathcal{B}$ receives $(\mathcal{I}, g_1, g_3, g_4, A_1A_2, B_2B_3)$ and T and simulates Game_{k-1} or Game_k with $\mathcal{A}$ depending on whether $T \in \mathbb{G}_{p_1p_3}$ or $T \in \mathbb{G}_{p_1p_2p_3}$.

$\mathcal{B}$ sets the public parameters by choosing random exponents $\alpha, a_1, \ldots, a_\ell, b, c \in \mathbb{Z}_N$ and setting $Y_1 = g_1, Y_3 = g_3, Y_4 = g_4, X_4 = Y_4^c, X_1 = Y_1^b$ and $u_i = Y_1^{a_i}$ for $i \in [\ell]$. $\mathcal{B}$ sends the public parameters $\mathsf{Pk} = (N, Y_1, Y_3, Y_4, t = X_1X_4, u_1, \ldots, u_\ell, \ \Omega = \mathbf{e}(Y_1, Y_1)^\alpha)$ to $\mathcal{A}$. Notice that $\mathcal{B}$ knows the master secret key $\mathsf{Msk} = (X_1, \alpha)$ associated with Pk. Let us now explain how $\mathcal{B}$ answers the i-th key query for identity $(\mathsf{ID}_{i,1}, \ldots, \mathsf{ID}_{i,j})$.

For $i < k$, $\mathcal{B}$ creates a semi-functional key by choosing random exponents $r_1, r_2, f, z, w \in \mathbb{Z}_N$ and, for $i \in \{1,2\}$, random $w_{i,2}, w_{i,j+1}, \ldots, w_{i,\ell} \in \mathbb{Z}_N$ and setting:

$$K_{1,1} = Y_1^{r_1} \cdot (B_2B_3)^f, \quad K_{1,2} = Y_1^\alpha \cdot (B_2B_3)^w \left(u_1^{\mathsf{ID}_{i,1}} \cdots u_j^{\mathsf{ID}_{i,j}} X_1\right)^{r_1} Y_3^{w_{1,2}},$$

$$E_{1,j+1} = u_{j+1}^{r_1} \cdot (B_2B_3)^{w_{1,j+1}}, \ldots, E_{1,\ell} = u_\ell^{r_1} \cdot (B_2B_3)^{w_{1,\ell}}.$$

and

$$K_{2,1} = Y_1^{r_2} \cdot (B_2B_3)^{zf}, \quad K_{2,2} = (B_2B_3)^{zw} \left(u_1^{\mathsf{ID}_{i,1}} \cdots u_j^{\mathsf{ID}_{i,j}} X_1\right)^{r_2} Y_3^{w_{2,2}},$$

$$E_{2,j+1} = u_{j+1}^{r_2} \cdot (B_2B_3)^{w_{2,j+1}}, \ldots, E_{2,\ell} = u_\ell^{r_2} \cdot (B_2B_3)^{w_{2,\ell}}.$$

By writing B_2 as g_2^ϕ, we have that this is a properly distributed semi-functional key with $\gamma = \phi \cdot f$ and $\gamma \cdot z_k = \phi \cdot w$.

For $i > k$, $\mathcal{B}$ runs the KeyGen algorithm using the master secret key $\mathsf{Msk} = (X_1, \alpha)$.

To answer the k-th key query for $\mathsf{ID}_k = (\mathsf{ID}_{k,1}, \ldots, \mathsf{ID}_{k,j})$, $\mathcal{B}$ sets $z_k = a_1\mathsf{ID}_{k,1} + \cdots + a_j\mathsf{ID}_{k,j} + b$, chooses random exponents $r'_2 \in \mathbb{Z}_N$ and, for $i \in \{1, 2\}$, random $w_{i,2}, w_{i,j+1}, \ldots, w_{i,\ell} \in \mathbb{Z}_N$, and sets:

$$K_{1,1} = T, \quad K_{1,2} = Y_1^{\alpha} \cdot T^{z_k} Y_3^{w_{1,2}}, \quad (E_{1,m} = T^{a_m} Y_3^{w_{1,m}})_{m=j+1}^{\ell}.$$

and

$$K_{2,1} = T^{r'_2}, \quad K_{2,2} = T^{r'_2 \cdot z_k} Y_3^{w_{2,2}}, \quad (E_{2,m} = T^{r'_2 \cdot a_m} Y_3^{w_{2,m}})_{m=j+1}^{\ell}.$$

We have the following two observations. If $T \in \mathbb{G}_{p_1p_3}$, then T can be written as $Y_1^{r'_1} Y_3^{r_3}$ and $(K_{i,1}, K_{i,2}, E_{i,j+1}, \ldots, E_{i,\ell})$ is a normal key with randomness $r_1 = r'_1$, $r_2 = r'_1 \cdot r'_2$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then T can be written as $Y_1^{r'_1} g_2^{s_2} Y_3^{r_3}$. In this case the key is a semi-functional key with randomness $r_1 = r'_1$, $r_2 = r'_1 \cdot r'_2$, $\gamma = s_2$ and $z = r'_2$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two pairs, $(M_0, \mathsf{ID}_0^\star = (\mathsf{ID}_{0,1}^\star, \ldots, \mathsf{ID}_{0,j}^\star))$ and $(M_1, \mathsf{ID}_1^\star = (\mathsf{ID}_{1,1}^\star, \ldots, \mathsf{ID}_{1,j}^\star))$. $\mathcal{B}$ chooses random $\beta \in \{0, 1\}$ and random $z, z' \in \mathbb{Z}_N$ and computes the challenge ciphertext as follows:

$$C_0 = M_\beta \cdot \mathbf{e}(A_1A_2, Y_1)^{\alpha}, \quad C_1 = (A_1A_2)^{a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b} Y_4^{z}, \quad C_2 = A_1A_2Y_4^{z'}.$$

This implicitly sets $Y_1^s = A_1$ and $z_c = a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b \pmod{p_2}$. Since ID_k is not a prefix of $\mathsf{ID}_\beta^\star$ modulo p_2, we have that z_k and z_c are independent and randomly distributed. We observe that, if $\mathcal{B}$ attempts to test whether the k-th key is semi-functional by using the above procedure to create a semi-functional ciphertext for ID_k, then we will have that $z_k = z_c$ and thus decryption always works (independently of T).

We can thus conclude that, if $T \in \mathbb{G}_{p_1p_3}$ then $\mathcal{B}$ has properly simulated Game_{k-1}. If $T \in \mathbb{G}_{p_1p_2p_3}$, then $\mathcal{B}$ has properly simulated Game_k. □

3.4.5 Indistinguishability of Game_q and $\mathsf{Game}_{\mathsf{Final}_0}$

Lemma 3.6 *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_q} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Final}_0}} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 2.

Proof. $\mathcal{B}$ receives $(\mathcal{I}, g_1, g_2, g_3, g_4, g_1^{\alpha}A_2, g_1^{s}B_2, g_2^{r}, A_2^{r})$ and T and simulates Game_q or $\mathsf{Game}_{\mathsf{Final}_0}$ with $\mathcal{A}$ depending on whether $T = \mathbf{e}(g_1, g_1)^{\alpha s}$ or T is a random element of $\mathbb{G}_T$.

$\mathcal{B}$ sets the public parameters as follows. $\mathcal{B}$ chooses random exponents $a_1, \ldots, a_\ell, b, c \in \mathbb{Z}_N$ and sets $Y_1 = g_1, Y_3 = g_3, Y_4 = g_4, X_4 = Y_4^c, X_1 = Y_1^b$, and $u_i = Y_1^{a_i}$ for $i \in [\ell]$. $\mathcal{B}$ computes $\Omega = \mathbf{e}(g_1^{\alpha}A_2, Y_1) = \mathbf{e}(Y_1^{\alpha}, Y_1)$ and send public parameters $\mathsf{Pk} = (N, Y_1, Y_2, Y_3, t = X_1X_4, u_1, \ldots, u_\ell, \Omega)$ to $\mathcal{A}$.

Each time $\mathcal{B}$ is asked to provide a key for an identity $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j)$, $\mathcal{B}$ creates a semi-functional key choosing random exponents $r_1, r_2, z, z' \in \mathbb{Z}_N$ and, for $i \in \{1, 2\}$, random $z_{i,j+1}, \ldots, z_{i,\ell}$, $w_{i,1}, w_{i,2}, w_{i,j+1}, \ldots, w_{i,\ell} \in \mathbb{Z}_N$ and setting:

$$K_{1,1} = Y_1^{r_1} \cdot g_2^{z} \cdot Y_3^{w_{1,1}}, \quad K_{1,2} = (g_1^{\alpha}A_2) \cdot g_2^{z'} \cdot \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1\right)^{r_1} \cdot Y_3^{w_{1,2}},$$

$$E_{1,j+1} = u_{j+1}^{r_1} \cdot g_2^{z_{1,j+1}} \cdot Y_3^{w_{1,j+1}}, \ \ldots, \ E_{1,\ell} = u_\ell^{r_1} \cdot g_2^{z_{1,\ell}} \cdot Y_3^{w_{1,\ell}}.$$

and

$$K_{2,1} = Y_1^{r_2} \cdot (g_2^{r})^{z} \cdot Y_3^{w_{2,1}}, \quad K_{2,2} = A_2^{r} \cdot (g_2^{r})^{z'} \cdot \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1\right)^{r_2} \cdot Y_3^{w_{2,2}},$$

$$E_{2,j+1} = u_{j+1}^{r_2} \cdot g_2^{z_{2,j+1}} \cdot Y_3^{w_{2,j+1}},\ \ldots,\ E_{2,\ell} = u_\ell^{r_2} \cdot g_2^{z_{2,\ell}} \cdot Y_3^{w_{2,\ell}}.$$

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two pairs, $(M_0, \mathsf{ID}_0^\star = (\mathsf{ID}_{0,1}^\star, \ldots, \mathsf{ID}_{0,j}^\star))$ and $(M_1, \mathsf{ID}_1^\star = (\mathsf{ID}_{1,1}^\star, \ldots, \mathsf{ID}_{1,j}^\star))$. $\mathcal{B}$ chooses random $\beta \in \{0,1\}$ and random $z, z' \in \mathbb{Z}_N$ and computes the challenge ciphertext as follows:

$$C_0 = M_\beta \cdot T, \quad C_1 = (g_1^s B_2)^{a_1 \mathsf{ID}_{\beta,1}^\star + \cdots + a_j \mathsf{ID}_{\beta,j}^\star + b} \cdot Y_4^z, \quad C_2 = g_1^s B_2 \cdot Y_4^{z'}.$$

This implicitly sets $z_c = (a_1 \mathsf{ID}_{\beta,1}^\star + \cdots + a_j \mathsf{ID}_{\beta,j}^\star + b) \bmod p_2$. We note that $u_i = Y_1^{a_i \bmod p_1}$ and $X_1 = Y_1^{b \bmod p_1}$ are elements of $\mathbb{G}_{p_1}$, so when $a_1, \cdots, a_\ell$ and b are randomly chosen from $\mathbb{Z}_N$, their value modulo p_1 and modulo p_2 are random and independent.

We finish by observing that, if $T = \mathbf{e}(g,g)^{\alpha s}$, then the ciphertext constructed is a properly distributed semi-functional ciphertext with message M_β. If T instead is a random element of $\mathbb{G}_T$, then the ciphertext is a semi-functional ciphertext with a random message. □

3.4.6 Indistinguishability of $\mathsf{Game}_{\mathsf{Final}_0}$ and $\mathsf{Game}_{\mathsf{Final}_1}$

Lemma 3.7 *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Final}_0}} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Final}_1}} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 3.

Proof. First, notice that if exists an adversary $\mathcal{A}'$ which distinguishes an encryption for an identity vector $\mathsf{ID}_0^\star$ from an encryption for an identity vector $\mathsf{ID}_1^\star$, where $\mathsf{ID}_0^\star$ and $\mathsf{ID}_1^\star$ are chosen by $\mathcal{A}'$, then there exists an adversary $\mathcal{A}$ which distinguishes an encryption for an identity $\mathsf{ID}^\star$ chosen by $\mathcal{A}$ from an encryption for a random identity vector. Hence, we suppose that we are simulating the games for a such adversary.

$\mathcal{B}$ receives $(\mathcal{I}, g_1, g_2, g_3, g_4, U, U^s A_{24}, U^{\hat{r}}, A_1 A_4, A_1^{\hat{r}} A_2, g_1^{\hat{r}} B_2, g_1^s B_{24})$ and T and simulates $\mathsf{Game}_{\mathsf{Final}_0}$ or $\mathsf{Game}_{\mathsf{Final}_1}$ with $\mathcal{A}$ depending on whether $T = A_1^s D_{24}$ or T is random in $\mathbb{G}_{p_1 p_2 p_4}$.

$\mathcal{B}$ sets the public parameters as follows. $\mathcal{B}$ chooses random exponents $\alpha, a_1, \ldots, a_\ell \in \mathbb{Z}_N$ and sets $Y_1 = g_1, Y_3 = g_3, Y_4 = g_4, t = A_1 A_4$, $u_i = U^{a_i}$ for $i \in [\ell]$, and $\Omega = \mathbf{e}(Y_1, Y_1)^\alpha$. $\mathcal{B}$ sends the public parameters $\mathsf{Pk} = (N, Y_1, Y_2, Y_3, t, u_1, \ldots, u_\ell, \Omega)$ to $\mathcal{A}$.

Each time $\mathcal{B}$ is asked to provide a key for an identity $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j)$, $\mathcal{B}$ creates a semi-functional key choosing random exponents $r_1', r_2' \in \mathbb{Z}_N$ and, for $\in \{1,2\}$, random $z_{i,j+1}, \ldots, z_{i,\ell}, w_{i,1}, w_{i,2}, w_{i,j+1}, \ldots, w_{i,\ell} \in \mathbb{Z}_N$ and setting:

$$K_{1,1} = (g_1^{\hat{r}} B_2)^{r_1'} Y_3^{w_{1,1}}, \quad K_{1,2} = Y_1^\alpha \left(\left(U^{\hat{r}}\right)^{a_1 \mathsf{ID}_1 + \cdots + a_j \mathsf{ID}_j} (A_1^{\hat{r}} A_2) \right)^{r_1'} Y_3^{w_{1,2}},$$

$$E_{1,j+1} = \left(U^{\hat{r}}\right)^{r_1' a_{j+1}} Y_2^{z_{1,j+1}} Y_3^{w_{1,j+1}}, \ldots, E_{1,\ell} = \left(U^{\hat{r}}\right)^{r_1' a_\ell} Y_2^{z_{1,\ell}} Y_3^{w_{1,\ell}}.$$

and

$$K_{2,1} = (g_1^{\hat{r}} B_2)^{r_2'} Y_3^{w_{2,1}}, \quad K_{2,2} = \left(\left(U^{\hat{r}}\right)^{a_1 \mathsf{ID}_1 + \cdots + a_j \mathsf{ID}_j} (A_1^{\hat{r}} A_2) \right)^{r_2'} Y_3^{w_{2,2}},$$

$$E_{2,j+1} = \left(U^{\hat{r}}\right)^{r_2' a_{j+1}} Y_2^{z_{2,j+1}} Y_3^{w_{2,j+1}}, \ldots, E_{2,\ell} = \left(U^{\hat{r}}\right)^{r_2' a_\ell} Y_2^{z_{2,\ell}} Y_3^{w_{2,\ell}}.$$

This implicitly sets the randomness $r_1 = \hat{r} r_1'$ and $r_2 = \hat{r} r_2'$. At some point, $\mathcal{A}$ sends $\mathcal{B}$ two pairs, $(M_0, \mathsf{ID}^\star = (\mathsf{ID}_1^\star, \ldots, \mathsf{ID}_j^\star))$ and $(M_1, \mathsf{ID}^\star = (\mathsf{ID}_1^\star, \ldots, \mathsf{ID}_j^\star))$. $\mathcal{B}$ chooses random $C_0 \in \mathbb{G}_T$ and computes the challenge ciphertext as follows:

$$C_0, \quad C_1 = T \left(U^s A_{24}\right)^{a_1 \mathsf{ID}_1^\star + \cdots + a_j \mathsf{ID}_j^\star}, \quad C_2 = g_1^s B_{24}.$$

This implicitly sets x and z_c to random values.

If $T = A_1^s D_{24}$, then this is properly distributed semi-functional ciphertext with C_0 random and for identity vector $\mathsf{ID}^\star$. If T is a random element of $\mathbb{G}_{p_1p_2p_4}$, then this is a semi-functional ciphertext with C_0 random in $\mathbb{G}_T$ and C_1 and C_2 random in $\mathbb{G}_{p_1p_2p_4}$.

Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

3.4.7 $\mathsf{Game}_{\mathsf{Final}_1}$ gives no advantage

Theorem 3.8 *If Assumptions* 1, 2 *and* 3 *hold then our Anonymous HIBE scheme is secure.*

Proof. If the assumptions hold then we have proved by the previous lemmata that the real security game is indistinguishable from $\mathsf{Game}_{\mathsf{Final}_1}$, in which the value of β is information-theoretically hidden from the attacker. Hence the attacker can obtain no advantage in breaking the Anonymous HIBE scheme. □

4 Secret-Key Anonymous IBE

4.1 Secret Key Identity Based Encryption

A Secret-Key Identity Based Encryption scheme (IBE) is a tuple of four efficient and probabilistic algorithms: (Setup, Encrypt, KeyGen, Decrypt).

$\mathsf{Setup}(1^\lambda)$: takes as input a security parameter λ and outputs the public parameters Pk and a master secret key Msk.

$\mathsf{KeyGen}(\mathsf{Msk}, \mathsf{ID})$: takes as input of the master secret key Msk, and an identity ID, and outputs a private key $\mathsf{Sk}_{\mathsf{ID}}$.

$\mathsf{Encrypt}(\mathsf{Msk}, M, \mathsf{ID})$: takes as input the master secret key Msk, a message M, and an identity ID and outputs a ciphertext Ct.

$\mathsf{Decrypt}(\mathsf{Ct}, \mathsf{Sk})$: takes as input a ciphertext Ct and a secret key Sk and outputs the message M, if the ciphertext was an encryption to an identity ID and the secret key is for the same identity.

4.2 Security definitions

We present the security of an Anonymous IBE scheme in secret key model. In this model, we have two definition of security: $c$iphertext security and $k$ey security.

4.2.1 Ciphertext Security definition

Security is defined through the following game, played by a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup. $\mathcal{C}$ runs the Setup algorithm to generate master secret key Msk which is kept secret.

Phase 1. $\mathcal{A}$ can make queries to the oracle Encrypt. To make a such query, $\mathcal{A}$ specifies a pair (M, ID) and receives an encryption of this pair computed using the Encrypt algorithm with Msk. $\mathcal{A}$ can make queries to the oracle KeyGen. To make a such query, $\mathcal{A}$ specifies an identity ID and receives a key of this identity computed using the KeyGen algorithm with Msk.

Challenge. $\mathcal{A}$ gives to $\mathcal{C}$ two pair message-identity (M_0, ID_0) and (M_1, ID_1). The identities must satisfy the property that no revealed identity in Phase 1 was either ID_0 or ID_1. $\mathcal{C}$ sets $\beta \in \{0, 1\}$ randomly and encrypts M_β under ID_β. $\mathcal{C}$ sends the ciphertext to the adversary.

Phase 2. This is the same as Phase 1 with the added restriction that any revealed identity must not be either ID_0 or ID_1.

Guess. $\mathcal{A}$ must output a guess β' for β. The advantage of $\mathcal{A}$ is defined to be $\mathrm{Prob}[\beta' = \beta] - \frac{1}{2}$.

Definition 4.1 *An Anonymous Identity Based Encryption scheme is* ciphertext-secure *if all polynomial time adversaries achieve at most a negligible (in λ) advantage in the previous security game.*

4.2.2 Key Security definition

Security is defined through the following game, played by a challenger $\mathcal{C}$ and an attacker $\mathcal{A}$.

Setup. $\mathcal{C}$ runs the Setup algorithm to generate master secret key Msk which is kept secret.

Phase 1. $\mathcal{A}$ can make queries to the oracle $\mathsf{Encrypt}$. To make a such query, $\mathcal{A}$ specifies a pair (M, ID) and receives an encryption of this pair computed using the $\mathsf{Encrypt}$ algorithm with the master secret key Msk. $\mathcal{A}$ can make queries to the oracle KeyGen. To make a such query, $\mathcal{A}$ specifies an identity ID and receives a key of this identity computed using the KeyGen algorithm with the master secret key Msk.

Challenge. $\mathcal{A}$ gives to $\mathcal{C}$ two identities ID_0 and ID_1. If in Phase 1 $\mathcal{A}$ did make a query (M, ID) to the oracle $\mathsf{Encrypt}$ such that ID was either ID_0 or ID_1, then the experiment fails. $\mathcal{C}$ sets $\beta \in \{0, 1\}$ randomly and compute the secret key for ID_β. $\mathcal{C}$ sends the secret key to the adversary.

Phase 2. This is the same as Phase 1 with the added restriction that if $\mathcal{A}$ did make a query (M, ID) to the oracle $\mathsf{Encrypt}$ such that ID was either ID_0 or ID_1, then the experiment fails.

Guess. $\mathcal{A}$ must output a guess β' for β. The advantage $\mathcal{A}$ is defined to be $\mathrm{Prob}[\beta' = \beta] - \frac{1}{2}$.

Definition 4.2 *A Secret-Key Anonymous Identity Based Encryption scheme is* key-secure *if all polynomial time adversaries achieve at most a negligible (in λ) advantage in the previous security game.*

Notice that no scheme with a deterministic KeyGen procedure can be key-secure.

4.3 Our construction

In this section we describe our construction for a Secret-key Anonymous IBE scheme which is similar to its public key version from the previous sections.

$\mathsf{Setup}(1^\lambda, 1^\ell)$: The setup algorithm chooses random description $\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ and random $Y_1, X_1, u \in \mathbb{G}_{p_1}, Y_3 \in \mathbb{G}_{p_3}, X_4, Y_4 \in \mathbb{G}_{p_4}$ and $\alpha \in \mathbb{Z}_N$. The *fictitious* public parameters are:

$$\mathsf{Pk} = (N, Y_1, Y_3, Y_4, t = X_1X_4, u, \Omega = \mathbf{e}(Y_1, Y_1)^\alpha).$$

The master secret key is $\mathsf{Msk} = (\mathsf{Pk}, X_1, \alpha)$.

$\mathsf{KeyGen}(\mathsf{Msk}, \mathsf{ID})$: The key generation algorithm chooses random $r \in \mathbb{Z}_N$ and also random elements $R_1, R_2 \in \mathbb{G}_{p_3}$ The secret key $\mathsf{Sk}_{\mathsf{ID}} = (K_1, K_2)$ is computed as

$$K_1 = Y_1^r R_1, \quad K_2 = Y_1^\alpha (u^{\mathsf{ID}} X_1)^r R_2.$$

$\mathsf{Encrypt}(\mathsf{Msk}, M, \mathsf{ID})$: The encryption algorithm chooses random $s \in \mathbb{Z}_N$ and random $Z, Z' \in \mathbb{G}_{p_4}$ The ciphertext (C_0, C_1, C_2) for the message $M \in \mathbb{G}_T$ is computed as

$$C_0 = M \cdot \mathbf{e}(Y_1, Y_1)^{\alpha s}, \quad C_1 = \left(u^{\mathsf{ID}} t\right)^s Z, \quad C_2 = Y_1^s Z'.$$

$\mathsf{Decrypt}(\mathsf{Msk}, \mathsf{Ct}, \mathsf{Sk})$: The decryption algorithm assumes that the key and ciphertext both correspond to the same identity ID. The decryption algorithm then computes the blinding factor similarly to the decryption procedure of the public-key version. Specifically,

$$\frac{\mathbf{e}(K_2, C_2)}{\mathbf{e}(K_1, C_1)} = \frac{\mathbf{e}(Y_1, Y_1)^{\alpha s} \mathbf{e}\left(u^{\mathsf{ID}} X_1, Y_1\right)^{rs}}{\mathbf{e}\left(Y_1, u^{\mathsf{ID}} X_1\right)^{rs}} = \mathbf{e}(Y_1, Y_1)^{\alpha s}.$$

4.4 Ciphertext Security

To prove ciphertext security of the Anonymous IBE scheme, we rely on the Assumptions $1, 2$ and 3 used in the proof of the public-key scheme.

We make the following considerations. If we instantiate the previous scheme as a public-key scheme by using the fictitious public-key parameter, it is identical to our public-key Anonymous IBE scheme (i.e., it is used in the non-hierarchical version). Thus, it is immediate to verify that from Assumptions $1, 2$ and 3 the security proof follows nearly identically. Generally, if a public-key IBE encryption scheme is semantically secure, its secret-key version is also semantically secure because we can simulate the encryption oracle by using the public-key. Therefore, we have the following theorem.

Theorem 4.3 *If Assumptions $1, 2$ and 3 hold, then our Secret-Key Anonymous IBE scheme is ciphertext-secure.*

4.5 Key Security

We will use *semi-functional ciphertexts* and *semi-functional keys* like defined previously. These will not be used in the real scheme, but we need them in our proofs. We include them for completeness.

Semi-functional Ciphertext. We let g_2 denote a generator of $\mathbb{G}_{p_2}$. A semi-functional ciphertext is created as follows: first, we use the encryption algorithm to form a normal ciphertext (C_0', C_1', C_2'). We choose random exponents $x, z_c \in \mathbb{Z}_N$. We set:

$$C_0 = C_0', \quad C_1 = C_1' g_2^{x z_c}, \quad C_2 = C_2' g_2^x.$$

Semi-functional Keys. To create a semi-functional key, we first create a normal key (K_1', K_2') using the key generation algorithm. We choose random exponents $\gamma, z_k \in \mathbb{Z}_N$. We set:

$$K_1 = K_1' g_2^{\gamma}, \quad K_2 = K_2' g_2^{\gamma z_k}.$$

We note that when a semi-functional key is used to decrypt a semi-functional ciphertext, the decryption algorithm will compute the blinding factor multiplied by the additional term $\mathbf{e}(g_2, g_2)^{x\gamma(z_k - z_c)}$. If $z_c = z_k$, decryption will still work. In this case, we say that the key is nominally semi-functional.

To prove the security of our scheme we rely on static Assumptions 1,2 and 3. For a PPT adversary $\mathcal{A}$ which makes q ciphertext queries, our proof of security will consist of the following $q + 3$ games between $\mathcal{A}$ and a challenger $\mathcal{C}$.

$\mathsf{Game}_{\mathsf{Real}}$: is the real key security game.

$\mathsf{Game}_{\mathsf{Restricted}}$: is the same as $\mathsf{Game}_{\mathsf{Real}}$ except that $\mathcal{A}$ cannot ask for keys for identities which are equal to one of the challenge identities modulo p_2. We will retain this restriction in all subsequent games.

Game_k: for k from 0 to q, Game_k is like $\mathsf{Game}_{\mathsf{Restricted}}$, except that the key given to $\mathcal{A}$ is semi-functional and the first k ciphertexts are semi-functional. The rest of the ciphertexts are normal.

$\mathsf{Game}_{\mathsf{Final}}$: is the same as Game_q, except that the challenge key is semi-functional with K_2 random in $\mathbb{G}_{p_1p_2p_4}$ (thus the key is independent from the identities provided by $\mathcal{A}$). It is clear that in this last game, no adversary can have advantage greater than 0.

We will show these games are indistinguishable in the following lemmata.

4.5.1 Indistinguishability of $\mathsf{Game}_{\mathsf{Real}}$ and $\mathsf{Game}_{\mathsf{Restricted}}$

Lemma 4.4 *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Real}}} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Restricted}}} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage $\geq \frac{\epsilon}{3}$ in breaking Assumption* 1.

Proof. The proof is identical to that given in lemma 3.3. □

4.5.2 Indistinguishability of $\mathsf{Game}_{\mathsf{Restricted}}$ and Game_0

Lemma 4.5 *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Restricted}}} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_0} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 1.

Proof. $\mathcal{B}$ receives $(\mathcal{I}, g_1, g_3, g_4, A_1A_2, B_2B_3)$ and T and simulates $\mathsf{Game}_{\mathsf{Restricted}}$ or Game_0 with $\mathcal{A}$ depending on whether $T \in \mathbb{G}_{p_1p_3}$ or $T \in \mathbb{G}_{p_1p_2p_3}$.

$\mathcal{B}$ sets the fictitious public parameters as follows. $\mathcal{B}$ chooses random exponents $\alpha, a, b, c \in \mathbb{Z}_N$ and sets $Y_1 = g_1, Y_3 = g_3, Y_4 = g_4$ $X_4 = Y_4^c, X_1 = Y_1^b$ and $u = Y_1^a$. $\mathcal{B}$ uses $\mathsf{Pk} = (N, Y_1, Y_3, Y_4, t = X_1X_4, u, \Omega = \mathbf{e}(Y_1, Y_1)^{\alpha})$ to respond to the ciphertext queries issued by $\mathcal{A}$. Notice that $\mathcal{B}$ knows also the master secret key $\mathsf{Msk} = (\mathsf{Pk}, X_1, \alpha)$ and thus can simulate all $\mathcal{A}$'s key queries.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two identities, $\mathsf{ID}_0^\star$ and $\mathsf{ID}_1^\star$. $\mathcal{B}$ chooses random $\beta \in \{0, 1\}$ and computes the challenge key as follows:

$$K_1 = T, \quad K_2 = Y_1^{\alpha} T^{a\mathsf{ID}_\beta^\star + b}.$$

We complete the proof with the following two observations. If $T \in \mathbb{G}_{p_1p_3}$, then T can be written as $Y_1^{s_1}Y_3^{s_3}$. In this case (K_1, K_2) is a normal key with randomness $r = s_1, R_1 = Y_3^{s_3}, R_2 = (Y_3^{s_3})^{a\mathsf{ID}^\star_\beta+b}$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then T can be written as $Y_1^{s_1}g_2^{s_2}Y_3^{s_3}$ and this case (K_1, K_2) is a semi-functional key with randomness $r = s_1, R_1 = Y_3^{s_3}, R_2 = (Y_3^{s_3})^{a\mathsf{ID}^\star_\beta+b}$, $\gamma = s_2$ and $z_c = a\mathsf{ID}^\star_\beta + b$. Thus, in the former case we have properly simulated $\mathsf{Game}_{\mathsf{Restricted}}$, and in the latter case we have simulated Game_0. □

4.5.3 Indistinguishability of Game_{k-1} and Game_k

Lemma 4.6 *Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{k-1}} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_k} = \epsilon$. Then, there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 1.

PROOF. $\mathcal{B}$ receives $(\mathcal{I}, g_1, g_3, g_4, A_1A_2, B_2B_3)$ and T and simulates Game_{k-1} or Game_k with $\mathcal{A}$ depending on whether $T \in \mathbb{G}_{p_1p_3}$ or $T \in \mathbb{G}_{p_1p_2p_3}$.

$\mathcal{B}$ sets the fictitious public parameters by choosing random exponents $\alpha, a, b, c \in \mathbb{Z}_N$ and setting $Y_1 = g_1, Y_3 = g_4, Y_4 = g_3, X_4 = Y_4^c, X_1 = Y_1^b$ and $u = Y_1^a$. Notice that $\mathcal{B}$ knows the master secret key $\mathsf{Msk} = (\mathsf{Pk}, X_1, \alpha)$ with $\mathsf{Pk} = (N, Y_1, Y_3, Y_4, t = X_1X_4, u, \Omega = \mathbf{e}(Y_1, Y_1)^\alpha)$ and thus can respond to all $\mathcal{A}$'s key queries. Let us now explain how $\mathcal{B}$ answers the i-th ciphertext query for pair (M, ID).

For $i < k$, $\mathcal{B}$ creates a semi-functional ciphertext by choosing random exponents $s, w_1, w_2 \in \mathbb{Z}_N$ and setting:

$$C_0 = M\mathbf{e}(Y_1, Y_1)^{\alpha s}, \quad C_1 = (u^{\mathsf{ID}}X_1)^s(B_2B_3)^{w_1}, \quad C_2 = Y_1^sY_4^{w_2}$$

By writing B_2 as g_2^ϕ, we have that this is a properly distributed semi-functional ciphertext with $x = \phi$ and $z_c = w_1$.

For $i > k$, $\mathcal{B}$ runs the $\mathsf{Encrypt}$ algorithm using the master secret key $\mathsf{Msk} = (\mathsf{Pk}, X_1, \alpha)$.

To answer the k-th ciphertext query for (M_k, ID_k), $\mathcal{B}$ sets $z_c = a\mathsf{ID}_k + b$, chooses random exponent $w_1, w_2 \in \mathbb{Z}_N$, and sets:

$$C_0 = M_k\mathbf{e}(T, Y_1)^\alpha, \quad C_1 = T^{z_c}Y_4^{w_1}, \quad C_2 = TY_4^{w_2}$$

We have the following two observations. If $T \in \mathbb{G}_{p_1p_3}$, then T can be written as $Y_1^{r_1}Y_4^{r_4}$ In this case this is a properly distributed normal ciphertext with $s = r_1$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then T can be written as $Y_1^{r_1}g_2^{r_2}Y_4^{r_4}$ and in this case it is a properly distributed semi-functional ciphertext with $x = r_2$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two identities, $\mathsf{ID}^\star_0$ and $\mathsf{ID}^\star_1$. $\mathcal{B}$ chooses random $\beta \in \{0, 1\}$ and random $z, z' \in \mathbb{Z}_N$ and computes the challenge key as follows:

$$K_1 = (A_1A_2)Y_3^z, \quad K_2 = Y_1^\alpha(A_1A_2)^{a\mathsf{ID}^\star_\beta+b}Y_3^{z'}$$

This implicitly sets $Y_1^r = A_1$ and $z_k = a\mathsf{ID}^\star_\beta + b \mod p_2$. Since ID_k is not equal to $\mathsf{ID}^\star_\beta$ modulo p_2, we have that z_k and z_c are independent and randomly distributed.

We can thus conclude that, if $T \in \mathbb{G}_{p_1p_3}$ then $\mathcal{B}$ has properly simulated Game_{k-1}. If $T \in \mathbb{G}_{p_1p_2p_3}$, then $\mathcal{B}$ has properly simulated Game_k. □

4.5.4 Indistinguishability of Game_q and $\mathsf{Game}_{\mathsf{Final}}$

Lemma 4.7 *Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_q} - \mathsf{Adv}^{\mathcal{A}}_{\mathsf{Game}_{\mathsf{Final}}} = \epsilon$. Then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 3.

Proof. First, notice that if exists an adversary $\mathcal{A}'$ which distinguishes an encryption for an identity $\mathsf{ID}_0^\star$ from an encryption for an identity $\mathsf{ID}_1^\star$, where $\mathsf{ID}_0^\star$ and $\mathsf{ID}_1^\star$ are chosen by $\mathcal{A}'$, then there exists an adversary $\mathcal{A}$ which distinguishes an encryption for an identity $\mathsf{ID}^\star$ chosen by $\mathcal{A}$ from an encryption for a random identity. Hence, we suppose that we are simulating the games for a such adversary.

$\mathcal{B}$ receives $(\mathcal{I}, g_1, g_2, g_3, g_4, U, U^s A_{24}, U^{\hat{r}}, A_1A_4, A_1^{\hat{r}}A_2, g_1^{\hat{r}}B_2, g_1^s B_{24})$ and T and simulates Game_q or $\mathsf{Game}_{\mathsf{Final}}$ with $\mathcal{A}$ depending on whether $T = A_1^s D_{24}$ or T is random in $\mathbb{G}_{p_1p_2p_4}$.

$\mathcal{B}$ chooses random exponents $\alpha \in \mathbb{Z}_N$ and sets $Y_1 = g_1, Y_3 = g_4, Y_4 = g_3$.

Each time $\mathcal{B}$ is asked to provide a ciphertext for an identity ID, $\mathcal{B}$ creates a semi-functional ciphertext choosing random exponents $r, w_1, w_2 \in \mathbb{Z}_N$ and sets

$$C_0 = M \cdot \mathbf{e}(g_1^{\hat{r}}B_2, Y_1)^{\alpha s},\ C_1 = (A_1^{\hat{r}}A_2)^{r\mathsf{ID}}(U^{\hat{r}})^r Y_4^{w_1},\ C_2 = (g_1^{\hat{r}}B_2)^r Y_4^{w_2}$$

This implicitly sets the randomness of the ciphertext to $\hat{r}r$, $u = A_1$ and $X_1 = U$.

Each time $\mathcal{B}$ is asked to provide a key for an identity ID, $\mathcal{B}$ creates a semi-functional key choosing random exponents $r, w_1, w_2 \in \mathbb{Z}_N$ and setting:

$$K_1 = (g_1^{\hat{r}}B_2)^r Y_3^{w_1}, \quad K_2 = Y_1^{\alpha}(A_1^{\hat{r}}A_2)^{r\mathsf{ID}}(U^{\hat{r}})^r Y_3^{w_2}.$$

This implicitly sets the randomness of the secret key to $\hat{r}r$.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ two identities, $\mathsf{ID}_0^\star$ and $\mathsf{ID}_1^\star$. $\mathcal{B}$ chooses random $w_1, w_2 \in \mathbb{Z}_N$ and computes the challenge secret key as follows:

$$K_1 = (g_1^s B_{24})Y_3^{w_1}, \quad K_2 = Y_1^{\alpha} T^{\mathsf{ID}_\beta^\star}(U^s A_{24})Y_3^{w_2}.$$

This implicitly sets γ and z_k to random values.

If $T = A_1^s D_{24}$, then this is properly distributed semi-functional key for identity $\mathsf{ID}_\beta^\star$. If T is a random element of $\mathbb{G}_{p_1p_2p_4}$, then this is a semi-functional key with K_2 random in $\mathbb{G}_{p_1p_2p_4}$.

Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

4.5.5 $\mathsf{Game}_{\mathsf{Final}}$ gives no advantage

Theorem 4.8 *If Assumptions* 1, 2 *and* 3 *hold then our Anonymous IBE scheme is both ciphertext and key secure.*

Proof. If the assumptions hold then we have proved by the previous lemmata that the real security game is indistinguishable from $\mathsf{Game}_{\mathsf{Final}}$, in which the value of β is information-theoretically hidden from the attacker. Hence the attacker can obtain no non-negligible advantage in breaking the key security of the Secret-key Anonymous IBE scheme. We have showed previously that it is also ciphertext-secure. □

5 Conclusions and Open Problems

We constructed the first Fully Secure Anonymous HIBE system with short ciphertexts in the public key model and the first fully secure IBE in the secret key model and proved their security in the standard model from simple and non-interactive assumptions generically secure. A drawback of our construction is that it uses bilinear groups of composite order. An open problem is to build such a scheme in symmetric bilinear groups of prime order. The general technique of Freeman [**?**] does not seem to apply to our scheme.

We also stress that our decryption algorithm works if the key and the ciphertext correspond to the same identity. It would be interesting to construct an anonymous HIBE in which the decryption algorithm works providef that the identity of the key is a prefix of the identity of the ciphertext.

To the best of our knowledge, Secret-Key Hierarchical IBE has not been studied before and we defer it to future work.

References

[1] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.

[2] Dan Boneh and Matthew K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[3] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341, Cambridge, MA, USA, February 10–12, 2005. Springer-Verlag, Berlin, Germany.

[4] Xavier Boyen. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography). In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–399, Santa Barbara, CA, USA, August 17–21, 2003. Springer-Verlag, Berlin, Germany.

[5] Xavier Boyen and Brent Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307, Santa Barbara, CA, USA, August 20–24, 2006. Springer-Verlag, Berlin, Germany.

[6] Angelo De Caro, Vincenzo Iovino, and Giuseppe Persiano. Fully Secure Anonymous HIBE and Secret-Key Anonymous IBE with Short Ciphertexts. Cryptology ePrint Archive, Report 2010/197, 2010. http://eprint.iacr.org/.

[7] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, Nice, France, May 10 –June 3, 2010. Springer-Verlag, Berlin, Germany. To appear.

[8] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *Cryptography and Coding, 8th IMA International Conference*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363, Cirencester, UK, December 17–19, 2001. Springer-Verlag, Berlin, Germany.

[9] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer-Verlag, Berlin, Germany.

[10] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption, 2010. http://eprint.iacr.org/2010/110.pdf, Eurocrypt 2010 to appear.

[11] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of*

Cryptography Conference, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479, Zurich, Switzerland, February 9–11, 2010. Springer-Verlag, Berlin, Germany.

[12] Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 215–234. Springer, 2009.

[13] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53, Santa Barbara, CA, USA, August 19–23, 1985. Springer-Verlag, Berlin, Germany.

[14] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 457–473, San Francisco, CA, USA, 2009. Springer-Verlag, Berlin, Germany.

[15] Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming: 35rd International Colloquium*, volume 5126 of *Lecture Notes in Computer Science*, pages 560–578, Reykjavik, Iceland, July 7–11, 2008. Springer-Verlag, Berlin, Germany.

[16] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636, Santa Barbara, CA, USA, August 16–20, 2009. Springer-Verlag, Berlin, Germany.

A Generic Security of Our Complexity Assumptions

We now prove that, if factoring is hard, our three complexity assumptions hold in the generic group model. We adopt the framework of [?] to reason about assumptions in bilinear groups $\mathbb{G}, \mathbb{G}_T$ of composite order $N = p_1p_2p_3p_4$. We fix generators $g_{p_1}, g_{p_2}, g_{p_3}, g_{p_4}$ of the subgroups $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}, \mathbb{G}_{p_4}$ and thus each element of $x \in \mathbb{G}$ can be expressed as $x = g_{p_1}^{a_1}, g_{p_2}^{a_2} g_{p_3}^{a_3} g_{p_4}^{a_4}$ for $a_i \in \mathbb{Z}_{p_i}$. For sake of ease of notation, we denote element $x \in \mathbb{G}$ by the tuple (a_1, a_2, a_3, a_4). We do the same with elements in $\mathbb{G}_T$ (with the respect to generator $\mathbf{e}(g_{p_i}, g_{p_i})$) and will denote elements in that group as bracketed tuples $[a_1, a_2, a_3, a_4]$. We use capital letters to denote random variables and reuse random variables to denote relationships between elements. For example, $X = (X_1, Y_1, Z_1, W_1)$ is a random element of $\mathbb{G}$, and $Y = (X_2, Y_1, Z_2, W_2)$ is another random element that shares the same $\mathbb{G}_{p_2}$ part.

We say that a random variable X is *dependent* from the random variables $\{A_i\}$ if there exists $\lambda_i \in \mathbb{Z}_N$ such that $X = \sum_i \lambda_i A_i$ as formal random variables. Otherwise, we say that X is *independent* of $\{A_i\}$. We state the following theorems from [?].

Theorem A.1 (Theorem A.1 of [?]) *Let $N = \prod_{i=1}^m p_i$ be a product of distinct primes, each greater than 2^λ. Let $\{A_i\}$ be random variables over $\mathbb{G}$ and $\{B_i\}, T_1$ and T_2 be random variables over $\mathbb{G}_T$. Denote by t the maximum degree of a random variable and consider the following experiment in the generic group model:*

Algorithm $\mathcal{A}$ is given $N, \{A_i\}, \{B_i\}$ and T_b for random $b \in \{0,1\}$ and outputs $b' \in \{0,1\}$. $\mathcal{A}$'s advantage is the absolute value of the difference between the probability that $b = b'$ and $1/2$.

Suppose that T_1 and T_2 are independent of $\{B_i\} \cup \{\mathbf{e}(A_i, A_j)\}$. Then if $\mathcal{A}$ performs at most q group operations and has advantage δ, then there exists an algorithm that outputs a nontrivial factor of N in time polynomial in λ and the running time of $\mathcal{A}$ with probability at least $\delta - \mathcal{O}(q^2 t/2^\lambda)$.

Theorem A.2 (Theorem A.2 of [?]) *Let $N = \prod_{i=1}^m p_i$ be a product of distinct primes, each greater than 2^λ. Let $\{A_i\}, T_1, T_2$ be random variables over $\mathbb{G}$ and let $\{B_i\}$ be random variables over G_T, where all random variables have degree at most t.*

Let $N = \prod_{i=1}^m p_i$ be a product of distinct primes, each greater than 2^λ. Let $\{A_i\}, T_1$ and T_2 be random variables over $\mathbb{G}$ and let $\{B_i\}$ be random variables over $\mathbb{G}_T$. Denote by t the maximum degree of a random variable and consider the same experiment as the previous theorem in the generic group model.

Let $S := \{i \mid \mathbf{e}(T_1, A_i) \neq \mathbf{e}(T_2, A_i)\}$ (where inequality refers to inequality as formal polynomials). Suppose each of T_1 and T_2 is independent of $\{A_i\}$ and furthermore that for all $k \in S$ it holds that $\mathbf{e}(T_1, A_k)$ is independent of $\{B_i\} \cup \{\mathbf{e}(A_i, A_j)\} \cup \{\mathbf{e}(T_1, A_i)\}_{i \neq k}$ and $\mathbf{e}(T_2, A_k)$ is independent of $\{B_i\} \cup \{\mathbf{e}(A_i, A_j)\} \cup \{\mathbf{e}(T_2, A_i)\}_{i \neq k}$. Then if there exists an algorithm $\mathcal{A}$ issuing at most q instructions and having advantage δ, then there exists an algorithm that outputs a nontrivial factor of N in time polynomial in λ and the running time of $\mathcal{A}$ with probability at least $\delta - \mathcal{O}(q^2 t/2^\lambda)$.

We apply these theorems to prove the security of our assumptions in the generic group model.

Assumption 1. We can express this assumption as:

$$A_1 = (1,0,0,0),\ A_2 = (0,0,1,0),\ A_3 = (0,0,0,1)$$

$$A_4 = (X_1, X_2, 0, 0),\ A_5 = (0, Y_2, Y_3, 0),$$

and

$$T_1 = (Z_1, Z_2, Z_3, 0), \quad T_2 = (Z_1, 0, Z_3, 0).$$

It is easy to see that T_1 and T_2 are both independent of $\{A_i\}$ because, for example, Z_1 does not appear in the A_i's. Next, we note that for this assumption we have $S = \{4, 5\}$, and thus, considering T_1 first, we obtain the following tuples:

$$C_{1,4} = \mathbf{e}(T_1, A_4) = [Z_1X_1, Z_2X_2, 0, 0], \quad C_{1,5} = \mathbf{e}(T_1, A_5) = [0, Z_2Y_2, Z_3Y_3, 0].$$

It is easy to see that $C_{1,k}$ with $k \in \{4, 5\}$ is independent of $\{\mathbf{e}(A_i, A_j)\} \cup \{\mathbf{e}(T_1, A_i)\}_{i \neq k}$. An analogous arguments apply for the case of T_2. Thus the independence requirements of Theorem A.2 are satisfied and Assumption 1 is generically secure, assuming it is hard to find a nontrivial factor of N.

Assumption 2. We can express this assumption as:

$$\begin{array}{lll} A_1 = (1, 0, 0, 0), & A_2 = (0, 1, 0, 0), & A_3 = (0, 0, 1, 0), \\ A_4 = (0, 0, 0, 1), & A_5 = (A, X_2, 0, 0), & A_6 = (S, Y_2, 0, 0) \\ A_7 = (0, X_2R, 0, 0), & A_8 = (0, R, 0, 0), & \end{array}$$

and

$$T_1 = [AS, 0, 0, 0], \quad T_2 = [Z_1, Z_2, Z_3, Z_4].$$

We note that Z_1 does not appear in $\{A_i\}$ and thus T_2 is independent from them. On the other hand, for T_1, the only way to obtain an element of $\mathbb{G}_T$ whose first component is AS is by computing $\mathbf{e}(A_5, A_6) = [AS, X_2Y_2, 0, 0]$ but there is no way to generate an element whose second component is X_2Y_2 and hence no way to cancel that term. Thus the independence requirement of Theorem A.1 is satisfied and Assumption 2 is generically secure, assuming it is hard to find a nontrivial factor of N.

Assumption 3. We can express this assumption as:

$$\begin{array}{llll} A_1 = (1, 0, 0, 0), & A_2 = (0, 1, 0, 0), & A_3 = (0, 0, 1, 0), & A_4 = (0, 0, 0, 1) \\ A_5 = (U, 0, 0, 0), & A_6 = (US, W_2, 0, W_4), & A_7 = (UR, 0, 0, 0), & A_8 = (X_1, 0, 0, X_4) \\ A_9 = (X_1R, X_2, 0, 0), & A_{10} = (R, Y_2, 0, 0), & A_{11} = (S, D_2, 0, Y_4), & \end{array}$$

and

$$T_1 = (X_1S, Z_2, 0, Z_4), \quad T_2 = (Z_1, Z_2, 0, Z_4).$$

It is easy to see that T_1 and T_2 are both independent of $\{A_i\}$ because, for example, Z_2 does not appear in the A_i's. Next we note that $S = \{1, 5, 6, 7, 8, 9, 10, 11\}$. Considering T_1 first, we obtain the following tuples:

$$\begin{array}{ll} C_{1,1} = \mathbf{e}(T_1, A_1) = [X_1S, 0, 0, 0], & C_{1,5} = \mathbf{e}(T_1, A_5) = [X_1SU, 0, 0, 0], \\ C_{1,6} = \mathbf{e}(T_1, A_6) = [X_1S^2U, Z_2W_2, 0, Z_4W_4], & C_{1,7} = \mathbf{e}(T_1, A_7) = [X_1SUR, 0, 0, 0], \\ C_{1,8} = \mathbf{e}(T_1, A_8) = [X_1^2S, 0, 0, Z_4X_4], & C_{1,9} = \mathbf{e}(T_1, A_9) = [X_1^2SR, Z_2X_2, 0, 0], \\ C_{1,10} = \mathbf{e}(T_1, A_{10}) = [X_1SR, Z_2Y_2, 0, 0], & C_{1,11} = \mathbf{e}(T_1, A_{11}) = [X_1S^2, Z_2D_2, 0, Z_4Y_4]. \end{array}$$

We start by observing that, for $k = 9, 10, 11$, $C_{1,k}$ is independent from $\{\mathbf{e}(A_i, A_j)\} \cup \{\mathbf{e}(T_1, A_i)\}_{i \neq k}$, since it is the only to contain Z_2X_2 for $k = 9$, Z_2Y_2 for $k = 10$, and Z_2D_2 for $k = 11$. Similarly, $C_{1,k}$

for $k = 6, 8$ is independent since it contains Z_4W_4, for $k = 6$, and Z_4X_4, for $k = 8$. Furthermore, for $C_{1,1}$, we observe that the only way to obtain an element whose first component contains X_1S is by computing $\mathbf{e}(A_8, A_{11}) = [X_1S, 0, 0, X_4Y_4]$ but then there is no way to generate an element whose fourth component is X_4Y_4 and hence no way to cancel that term. Similarly for $C_{1,5}$ and $C_{1,7}$. To obtain an element whose first component contains X_1SU (resp. X_1SUR) the only way is by computing $\mathbf{e}(A_8, A_6) = [X_1US, 0, 0, X_4W_4]$ (rasp. $\mathbf{e}(A_6, A_9) = [USX_1R, X_2W_2, 0, 0]$) but there is no way to cancel the fourth (resp. second) component X_4W_4 (resp. X_2W_2).

Analogous arguments apply for the case of T_2.

Thus the independence requirement of Theorem A.2 is satisfied and Assumption 3 is generically secure, assuming it is hard to find a nontrivial factor of N.

Assumption 4. We can express this assumption as:

$$\begin{array}{llll} A_1 = (1, 0, 0, 0), & A_2 = (0, 1, 0, 0), & A_3 = (0, 0, 1, 0), & A_4 = (0, 0, 0, 1) \\ A_5 = (U, M_2, 0, 0), & A_6 = (UR, N_2, 0, 0), & A_7 = (X_1, O_2, 0, 0), & A_8 = (X_1R, P_2, 0, 0) \\ A_9 = (R, Q_2, 0, 0), & A_{10} = (S, Y_2, 0, 0), & A_{11} = (X_1S, W_2, 0, 0), & \end{array}$$

and

$$T_1 = (US, Z_2, 0, 0), \quad T_2 = (Z_1, Z_2, 0, 0).$$

It is easy to see that T_1 and T_2 are both independent of $\{A_i\}$ because, for example, Z_2 does not appear in the A_i's. Next we note that $S = \{1, 5, 6, 7, 8, 9, 10, 11\}$. Considering T_1 first, we obtain the following tuples:

$$\begin{array}{ll} C_{1,1} = \mathbf{e}(T_1, A_1) = [US, 0, 0, 0], & C_{1,5} = \mathbf{e}(T_1, A_5) = [U^2S, Z_2M_2, 0, 0], \\ C_{1,6} = \mathbf{e}(T_1, A_6) = [U^2SR, Z_2N_2, 0, 0], & C_{1,7} = \mathbf{e}(T_1, A_7) = [USX_1, Z_2O_2, 0, 0], \\ C_{1,8} = \mathbf{e}(T_1, A_8) = [USX_1R, Z_2P_2, 0, 0], & C_{1,9} = \mathbf{e}(T_1, A_9) = [USR, Z_2Q_2, 0, 0], \\ C_{1,10} = \mathbf{e}(T_1, A_{10}) = [US^2, Z_2Y_2, 0, 0], & C_{1,11} = \mathbf{e}(T_1, A_{11}) = [US^2X_1, Z_2W_2, 0, 0]. \end{array}$$

It is easy to see that $C_{1,1}$ is independent from $\{\mathbf{e}(A_i, A_j)\} \cup \{\mathbf{e}(T_1, A_i)\}_{i \neq 1}$, since there is no way to obtain an element whose first component contains US. Furthermore, for $k = 5, 6, 7, 8, 9, 10, 11$, $C_{1,k}$ is independent from $\{\mathbf{e}(A_i, A_j)\} \cup \{\mathbf{e}(T_1, A_i)\}_{i \neq k}$, since it is the only to contain the particular $\mathbb{G}_{p_2}$ part.

Analogous arguments apply for the case of T_2.

Thus the independence requirement of Theorem A.2 is satisfied and Assumption 5 is generically secure, assuming it is hard to find a nontrivial factor of N.