Fully Secure Anonymous HIBE with Short Ciphertexts

Angelo De Caro Vincenzo Iovino*
Giuseppe Persiano

Dipartimento di Informatica ed Applicazioni,
Università di Salerno, 84084 Fisciano (SA), Italy.
{decaro,iovino,giuper}@dia.unisa.it.

Friday 9^{th} April, 2010

Abstract

In [LW10], Lewko and Waters 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 groups (of four primes).

1 Introduction

Identity-Based Encryption (IBE) was introduced by [Sha85] 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 [BF03] which makes use of bilinear groups and to [Coc01] which uses quadratic residues. Later, [HL02] 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. Following these works, it followed interest in Anonymous IBE, where the ciphertext does not leak the identity of the recipient. Such systems enjoy a very useful privacy mechanism of privacy and can be used to make search over encrypted data. Interpreting the identities as keywords, Anonymous IBE allows the encryptor to make the document searchable by keywords, where the capabilities to search on particular keywords are delegated by a central authority. Anonymous IBE can be used to build Public-key Encryption with Keyword Search [BDOP04]. As noticed by [Boy03], the first solution to Anonymous IBE was implicit in the paper of [BF03] though the authors did not state it explicitly. The drawback of the IBE of [BF03] is that its security proof uses the random oracle model. [CHK03] introduced a weaker notion of

*Work done while visiting the Department of Computer Science of The Johns Hopkins University.

security called selective-ID, where the attacker choose the identity to attack before it receives the public parameters. In this model [BW06] described an Anonymous Hierarchical Identity-Based Encryption system in the standard model. The first efficient IBE system with full security (non selective-ID) in the standard model was described by [Wat05]. [GH09] described a fully secure HIBE system, although this system is based on a complicated assumption and security proof. [BBG05] constructed an HIBE system with short ciphertexts in the selective-ID model. [Wat09] introduced a proof methodology called Dual System Encryption to prove the full-security of (H)IBE systems. His construction of HIBE is based on simple and established Decision Linear assumption. Recently, [LW10] use the previous methodology to construct the first fully secure HIBE system with short ciphertexts improving the previous result of [BBG05]. The drawback of the latter construction is that it is inherently non anonymous. [SKOS09] build an Anonymous HIBE but their security proof is in the selective-ID model. We show that with an immediate modification to the HIBE of [LW10], we can achieve the first fully secure Anonymous HIBE with short ciphertexts. Recently [LOS+10] built a fully-secure hierarchical predicate encryption system which has as special case Anonymous HIBE, but it has non-constant size ciphertexts and keys are larger than in our construction resulting in a less efficient scheme when instantiated as HIBE. In [CHKP10] the authors constructed the first Anonymous HIBE scheme based on hard lattice problems but the size of a ciphertext depends on the depth of the hierarchy.

2 Model and security notions

2.1 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, $\mathsf{Encrypt}$, KeyGen, $\mathsf{Decrypt}$, $\mathsf{Delegate}$).

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

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

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

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

$\mathsf{Decrypt}(\mathsf{Pk}, \mathsf{Ct}, \mathsf{Sk})$: 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 $\mathsf{Encrypt}$ algorithm on public parameters Pk, message M and identity ID and that Sk is a secret key identity ID obtained through a sequence of KeyGen and $\mathsf{Delegate}$ calls using the same public parameters Pk. Then $\mathsf{Decrypt}$ returns M except with negligible probability.

2.2 Security definition

We give complete form of the security definition following [SW08]. 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, the $\mathcal{C}$ creates a key for this vector by calling the key generation algorithm, and places this key in the set S. It 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, the $\mathcal{C}$ 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}. It 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 to $\mathcal{C}$ two pair message-identity $(M_0, \mathsf{ID}_0^\star)$ and $(M_1, \mathsf{ID}_1^\star)$. The identity vector must satisfy the property that no revealed identity in Phase 1 was a prefix of either $\mathsf{ID}_0^\star$ or $\mathsf{ID}_1^\star$. $\mathcal{A}$ chooses random $\beta \in \{0, 1\}$ and encrypts M_β under $\mathsf{ID}_\beta^\star$. $\mathcal{C}$ sends the ciphertext to the adversary.

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

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

Definition 2.1 *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 Composite Order Bilinear Groups

Composite order bilinear groups were first used in cryptographic construction in [BGN05]. We use groups of order product of four primes and a generator $\mathcal{G}$ which takes as input security parameter λ and outputs and 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 $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ 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.

3.1 Hardness Assumptions

For a generator $\mathcal{G}$, we define the following distribution:

$$\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e}) \leftarrow \mathcal{G}(1^\lambda),$$

$$g_1, A_1 \leftarrow \mathbb{G}_{p_1}, A_2, B_2 \leftarrow \mathbb{G}_{p_2}, g_3, B_3 \leftarrow \mathbb{G}_{p_3}, g_4 \leftarrow \mathbb{G}_{p_4},$$
$$D = (\mathcal{I}, g_1, g_3, g_4, A_1A_2, B_2B_3),$$
$$T_1 \leftarrow \mathbb{G}_{p_1p_2p_3}, \quad T_2 \leftarrow \mathbb{G}_{p_1p_3}.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be:

$$\mathsf{Adv1}_{\mathcal{A}}(\lambda) = |\mathrm{Prob}[\mathcal{A}(D, T_1) = 1] - \mathrm{Prob}[\mathcal{A}(D, T_2) = 1]|.$$

Assumption 1 *We say that Assumption 1 holds for generator $\mathcal{G}$ if for all probabilistic polynomial-time algorithms $\mathcal{A}$ $\mathsf{Adv1}_{\mathcal{A}}(1^\lambda)$ is a negligible function of λ.*

For a generator $\mathcal{G}$, we define the following distribution:

$$\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e}) \leftarrow \mathcal{G}(1^\lambda),$$

$$\alpha, s \leftarrow \mathbb{Z}_N,$$
$$g_1 \leftarrow \mathbb{G}_{p_1}, g_2, A_2, B_2 \leftarrow \mathbb{G}_{p_2}, g_3 \leftarrow \mathbb{G}_{p_3}, g_4 \leftarrow \mathbb{G}_{p_4},$$
$$D = (\mathcal{I}, g_1, g_2, g_3, g_4, g_1^\alpha A_2, g_1^s B_2),$$
$$T_1 = \mathbf{e}(g_1, g_1)^{\alpha s}, \quad T_2 \leftarrow \mathbb{G}_T.$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be:

$$\mathsf{Adv2}_{\mathcal{A}}(1^\lambda) = |\mathrm{Prob}[\mathcal{A}(D, T_1) = 1] - \mathrm{Prob}[\mathcal{A}(D, T_2) = 1]|.$$

Assumption 2 *We say that Assumption 2 holds for generator $\mathcal{G}$ if for all probabilistic polynomial time algorithm $\mathcal{A}$ $\mathsf{Adv2}_{\mathcal{A}}(1^\lambda)$ is a negligible function of λ.*

For a generator $\mathcal{G}$, we define the following distribution:

$$\mathcal{I} = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \mathbf{e}) \leftarrow \mathcal{G}(1^\lambda),$$

$$\hat{r}, s \leftarrow \mathbb{Z}_N,$$

$$g_1, U, A_1 \leftarrow \mathbb{G}_{p_1}, g_2, A_2, B_2, D_2, F_2 \leftarrow \mathbb{G}_{p_2}, g_3 \leftarrow \mathbb{G}_{p_3}, g_4, A_4, B_4, D_4 \leftarrow \mathbb{G}_{p_4},$$

$$A_{24}, B_{24}, D_{24} \leftarrow \mathbb{G}_{p_2p_4},$$

$$D = (\mathcal{I}, g_1, g_2, g_3, g_4, U, U^s A_{24}, U^{\hat{r}}, A_1A_4, A_1^{\hat{r}}A_2, g_1^{\hat{r}}B_2, g_1^s B_{24}),$$

$$T_1 = A_1^s D_{24}, \quad T_2 \leftarrow \mathbb{G}_{p_1p_2p_4}$$

We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be:

$$\mathsf{Adv3}_{\mathcal{A}}(1^\lambda) = |\mathrm{Prob}[\mathcal{A}(D, T_1) = 1] - \mathrm{Prob}[\mathcal{A}(D, T_2) = 1]|.$$

Assumption 3 *We say that Assumption 3 holds for generator $\mathcal{G}$ if for all probabilistic polynomial time algorithm $\mathcal{A}$ $\mathsf{Adv3}_{\mathcal{A}}(1^\lambda)$ is a negligible function of λ.*

4 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 $r \in \mathbb{Z}_N$ and also random elements $R_1, R_2, R_{j+1}, \ldots, R_\ell \in \mathbb{G}_{p_3}$ (this is done by raising Y_3 to a random power). The secret key $\mathsf{Sk}_{\mathsf{ID}} = (K_1, K_2, E_{j+1}, \ldots, E_\ell)$ is computed as

$$K_1 = Y_1^r R_1, \quad K_2 = Y_1^\alpha \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1\right)^r R_2, \quad E_{j+1} = u_{j+1}^r R_{j+1}, \ \ldots, \ E_\ell = u_\ell^r R_\ell.$$

$\mathsf{Delegate}(\mathsf{Pk}, \mathsf{ID}, \mathsf{Sk}_{\mathsf{ID}}, \mathsf{ID}_{j+1})$: Given a key $\mathsf{Sk}_{\mathsf{ID}} = (K_1', K_2', E_{j+1}', \ldots, E_\ell')$ 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 $r \in \mathbb{Z}_N$ and random $R_1, R_2, R_{j+2}, \ldots, R_\ell \in \mathbb{G}_{p_3}$. The secret key $(K_1, K_2, E_{j+1}, \ldots, E_\ell)$ is computed as

$$K_1 = K_1' Y_1^r R_1, \quad K_2 = K_2' \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1\right)^r \left(E_{j+1}'\right)^{\mathsf{ID}_{j+1}} u_{j+1}^{r\mathsf{ID}_{j+1}} R_2,$$

$$E_{j+2} = E_{j+2}' u_{j+2}^r R_{j+2}, \ldots, E_\ell = E_\ell' u_\ell^r R_\ell.$$

We observe that the new key has the same distributions as the key computed by the KeyGen algorithm on $(\mathsf{ID}_1, \ldots, \mathsf{ID}_j, \mathsf{ID}_{j+1})$.

$\mathsf{Encrypt}(\mathsf{Pk}, M, \mathsf{ID} = (\mathsf{ID}_1, \ldots, \mathsf{ID}_j))$: The encryption algorithm chooses random $s \in \mathbb{Z}_N$ and random $Z, Z' \in \mathbb{G}_{p_4}$ (this is done by raising Y_4 to a random power). 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:

$$\frac{\mathbf{e}(K_2, C_2)}{\mathbf{e}(K_1, C_1)} = \frac{\mathbf{e}(Y_1, Y_1)^{\alpha s} \mathbf{e}\left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1, Y_1\right)^{rs}}{\mathbf{e}\left(Y_1, u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1\right)^{rs}} = \mathbf{e}(Y_1, Y_1)^{\alpha s}.$$

5 Security

To prove security of our Anonymous HIBE scheme, we rely on the static Assumptions 1, 2 and 3. Following Lewko and Waters [LW10], 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 $(K_1', K_2', E_{j+1}', \ldots, E_\ell')$ using the key generation algorithm. We choose random exponents $\gamma, z_k, z_{j+1}, \ldots, z_\ell \in \mathbb{Z}_N$. We set:

$$K_1 = K_1' g_2^\gamma, \quad K_2 = K_2' g_2^{\gamma z_k}, \quad E_{j+1} = E_{j+1}' g_2^{\gamma z_{j+1}}, \ \ldots, \ E_\ell = E_\ell' g_2^{\gamma z_\ell}.$$

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.

Our proof of security will be structured as a hybrid argument over a sequence of games. The first game, $\mathsf{Game}_{\mathsf{Real}}$, is the real Anonymous HIBE security game. The next 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, (the challenger will not be asked to delegate keys in a particular way). The next game, $\mathsf{Game}_{\mathsf{Restricted}}$ is the same as $\mathsf{Game}_{\mathsf{Real}'}$ except that the adversary 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. We let q denote the number of key queries the attacker makes. For k from 0 to q, we define Game_k like $\mathsf{Game}_{\mathsf{Restricted}}$, except that the ciphertext given to the attacker is semi-functional and the first k keys are semi-functional. The rest of the keys are normal.

We define $\mathsf{Game}_{\mathsf{Final}_0}$ to be like Game_q, except that the challenge ciphertext is a semi-functional encryption of a random message, not one of the messages provided by the attacker. Furthermore, we define $\mathsf{Game}_{\mathsf{Final}_1}$ to be like $\mathsf{Game}_{\mathsf{Final}_0}$, except that the challenge ciphertext is a semi-functional encryption for a random identity, not one of the identities provided by the attacker. It is clear that in this last game, no adversary can have non-negligible advantage.

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

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

Lemma 5.1 *For any algorithm $\mathcal{A}$, $\mathsf{Game}_{\mathsf{Real}}\mathsf{Adv}_{\mathcal{A}} = \mathsf{Game}_{\mathsf{Real}'}\mathsf{Adv}_{\mathcal{A}}$.*

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. □

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

Lemma 5.2 *Suppose that there exists an algorithm $\mathcal{A}$ such that $\mathsf{Game}_{\mathsf{Real}'}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{\mathsf{Restricted}}\mathsf{Adv}_{\mathcal{A}} = \epsilon$. Then there exists a probabilistic polynomial-time 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 $\mathsf{ID}^\star = (\mathsf{ID}_1^\star, \ldots, \mathsf{ID}_j^\star)$ modulo p_2. That is, there exists i and $j \in \{0,1\}$ such that that $\mathsf{ID}_i \neq \mathsf{ID}_{j,i}^\star$ modulo N and that p_2 divides $\mathsf{ID}_i - \mathsf{ID}_{j,i}^\star$ and thus $a = \gcd(\mathsf{ID}_i - \mathsf{ID}_{j,i}^\star, N)$ 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 $\mathsf{ID}^\star \in \{\mathsf{ID}_0^\star, \mathsf{ID}_1^\star\}$, $\mathcal{B}$ computes $a = \gcd(\mathsf{ID}_i - \mathsf{ID}_i^\star, N)$. 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 $\mathbf{e}(T^a, A_1A_2)$ 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). $\square$

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

Lemma 5.3 *Suppose that there exists an algorithm $\mathcal{A}$ such that $\mathsf{Game}_{\mathsf{Restricted}}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_0\mathsf{Adv}_{\mathcal{A}} = \epsilon$. Then there exists a probabilistic polynomial-time 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 $Y_1^{s_1}Y_3^{s_3}$. In this case (C_0, C_1, C_2) is a normal ciphertext with randomness $s = s_1, Z = Y_3^{s_3a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b}$ and $Z' = Y_3^{s_3}$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then T can be written as $Y_1^{s_1}g_2^{s_2}Y_3^{s_3}$ and this case (C_0, C_1, C_2) is a semi-functional ciphertext with randomness $s = s_1, Z = Y_3^{s_3a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b}$, $Z' = Y_3^{s_3}$, $\gamma = s_2$ and $z_c = a_1\mathsf{ID}_{\beta,1}^\star + \cdots + a_j\mathsf{ID}_{\beta,j}^\star + b$. $\square$

5.4 Indistinguishability of Game_{k-1} and Game_k

Lemma 5.4 *Suppose there exists an algorithm $\mathcal{A}$ such that $\mathsf{Game}_{k-1}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_k\mathsf{Adv}_{\mathcal{A}} = \epsilon$. Then, there exists a probabilistic polynomial-time 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 $r, f, w, w_{j+1}, \ldots, w_\ell \in \mathbb{Z}_N$ and setting:

$$K_1 = Y_1^r(B_2B_3)^f, \quad K_2 = Y_1^\alpha\left(u_1^{\mathsf{ID}_{i,1}} \cdots u_j^{\mathsf{ID}_{i,j}} X_1\right)^r (B_2B_3)^w,$$

$$E_{j+1} = u_{j+1}^r (B_2 B_3)^{w_{j+1}}, \ldots, E_\ell = u_\ell^r (B_2 B_3)^{w_\ell}.$$

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 $w_k, w_{j+1}, \ldots, w_\ell \in \mathbb{Z}_N$, and sets:

$$K_1 = T, \quad K_2 = Y_1^{\alpha} T^{z_k} Y_3^{w_k}, \quad E_{j+1} = T^{a_{j+1}} Y_3^{w_{j+1}}, \ \ldots, \ E_\ell = T^{a_\ell} Y_3^{w_\ell}.$$

We have the following two observations. If $T \in \mathbb{G}_{p_1 p_3}$, then T can be written as $Y_1^{r_1} Y_3^{r_3}$ and $(K_1, K_2, E_{j+1}, \ldots, E_\ell)$ is a normal key with randomness $r = r_1$, $R_1 = Y_3^{s_3}$, $R_2 = Y_3^{s_3 z_k} Y_3^{w_k}$, $R_{j+1} = Y_3^{w_{j+1}}$ and $R_\ell = Y_3^{w_\ell}$. If $T \in \mathbb{G}_{p_1 p_2 p_3}$, 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 $r = r_1$, $R_1 = Y_3^{s_3}$, $R_2 = Y_3^{s_3 z_k} Y_3^{w_k}$, $R_{j+1} = Y_3^{w_{j+1}}$, $R_\ell = Y_3^{w_\ell}$, $\gamma = s_2$.

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:

$$C_0 = M_\beta \cdot \mathbf{e}(A_1 A_2, Y_1)^{\alpha}, \quad C_1 = (A_1 A_2)^{a_1 \mathsf{ID}_{\beta,1}^\star + \cdots + a_j \mathsf{ID}_{\beta,j}^\star + b} Y_4^{z}, \quad C_2 = A_1 A_2 Y_4^{z'}.$$

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 itself whether key k 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 $T \in \mathbb{G}_{p_1 p_3}$ then $\mathcal{B}$ has properly simulated Game_{k-1}. If $T \in \mathbb{G}$, then $\mathcal{B}$ has properly simulated Game_k. □

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

Lemma 5.5 *Suppose that there exists an algorithm $\mathcal{A}$ such that* $\mathsf{Game}_q\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{\mathsf{Final}_0}\mathsf{Adv}_{\mathcal{A}} = \epsilon$. *Then there exists a probabilistic polynomial-time algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption* 2.

Proof. $\mathcal{B}$ receives $(\mathcal{I}, g_1, g_2, g_3, g_4, g_1^{\alpha} A_2, g_1^s B_2)$ 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 $T \in \mathbb{G}_T$ random.

$\mathcal{B}$ 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 $\mathsf{Pk} = (N, Y_1, Y_2, Y_3, t = X_1 X_4, u_1, \ldots, u_\ell, \Omega)$ 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 $r, f, w, z, z', z_{j+1}, \ldots, z_\ell, w, w_{j+1}, \ldots, w_\ell \in \mathbb{Z}_N$ and setting:

$$K_1 = Y_1^r Y_3^f g_2^z, \quad K_2 = (g_1^{\alpha} A_2) \cdot g_2^{z'} \cdot \left(u_1^{\mathsf{ID}_1} \cdots u_j^{\mathsf{ID}_j} X_1\right)^r \cdot Y_3^w, \quad E_{j+1} = u_{j+1}^r Y_3^{w_{j+1}} g_2^{z_{j+1}}, \ \ldots, \ E_\ell = u_\ell^r Y_3^{w_\ell} g_2^{z_\ell}.$$

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:

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

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. □

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

Lemma 5.6 *Suppose that there exists an algorithm $\mathcal{A}$ such that $\mathsf{Game}_{\mathsf{Final}_0}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{\mathsf{Final}_1}\mathsf{Adv}_{\mathcal{A}} = \epsilon$. Then there exists a probabilistic polynomial-time 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 $T = A_1^s A_{24}$ or T is random in $\mathbb{G}_{p_1 p_2 p_4}$ random.

$\mathcal{B}$ 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 $\tilde{r}, f, c, w, w_{j+1}, \ldots, w_\ell, z_{j+1}, \ldots, z_\ell \in \mathbb{Z}_N$ and setting:

$$K_1 = (g_1^{\hat{r}} B_2)^{\tilde{r}} Y_3^f, \quad K_2 = Y_1^\alpha \left(\left(U^{\hat{r}}\right)^{a_1 \mathsf{ID}_1 + \cdots + a_j \mathsf{ID}_j} (A_1^{\hat{r}} A_2) \right)^{\tilde{r}} Y_3^w,$$

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

This implicitly sets the randomness $r = \hat{r}\tilde{r}$. 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:

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

This implicitly sets x and z_c to random values.

If $T = A_1^s B_{24}$, 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 $\mathbb{G}_{p_1 p_2 p_4}$, then this is a semi-functional ciphertext with C_0 random and for random identity. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between these possibilities for T. □

5.7 Main Theorem

Theorem 5.7 *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 $Game_{Final_1}$, 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. $\square$

6 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 [KSW08] 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 [KSW08].

Theorem 6.1 (Theorem A.1 of [KSW08]) *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* $\delta - \mathcal{O}(q^2t/2^\lambda)$.

Theorem 6.2 (Theorem A.2 of [KSW08]) *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 $\delta - \mathcal{O}(q^2t/2^\lambda)$.

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

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

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 6.2 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:

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

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 6.1 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:

$$
\begin{array}{ll}
C_{1,1} = \mathbf{e}(T_1, A_1) = [X_1 S, 0, 0, 0], & C_{1,5} = \mathbf{e}(T_1, A_5) = [X_1 SU, 0, 0, 0], \\
C_{1,6} = \mathbf{e}(T_1, A_6) = [X_1 S^2 U, Z_2 W_2, 0, Z_4 W_4], & C_{1,7} = \mathbf{e}(T_1, A_7) = [X_1 SUR, 0, 0, 0], \\
C_{1,8} = \mathbf{e}(T_1, A_8) = [X_1^2 S, 0, 0, Z_4 X_4], & C_{1,9} = \mathbf{e}(T_1, A_9) = [X_1^2 SR, Z_2 X_2, 0, 0], \\
C_{1,10} = \mathbf{e}(T_1, A_{10}) = [X_1 SR, Z_2 Y_2, 0, 0], & C_{1,11} = \mathbf{e}(T_1, A_{11}) = [X_1 S^2, Z_2 D_2, 0, Z_4 Y_4].
\end{array}
$$

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 $Z_2 X_2$ for $k = 9$, $Z_2 Y_2$ for $k = 10$, and $Z_2 D_2$ for $k = 11$. Similarly, $C_{1,k}$ for $k = 6, 8$ is independent since it contains $Z_4 W_4$, for $k = 6$, and $Z_4 X_4$, for $k = 8$. Furthermore, for $C_{1,1}$, we observe the the only way to obtain an element whose first component contains $X_1 S$ is by computing $\mathbf{e}(A_8, A_{11}) = [X_1 S, 0, 0, X_4 Y_4]$ but then there is no way to generate an element whose fourth component is $X_4 Y_4$ 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 $X_1 SU$ (resp. $X_1 SUR$) the only way is by computing $\mathbf{e}(A_8, A_6) = [X_1 US, 0, 0, X_4 W_4]$ (rasp. $\mathbf{e}(A_6, A_9) = [USX_1 R, X_2 W_2, 0, 0]$) but there is no way to cancel the fourth (resp. second) component $X_4 W_4$ (resp. $X_2 W_2$).

Analogous arguments apply for the case of T_2.

Thus the independence requirement of Theorem 6.2 is satisfied and Assumption 3 is generically secure, assuming it is hard to find a nontrivial factor of N.

7 Conclusions and Open Problems

We constructed the first Fully Secure Anonymous HIBE system with short ciphertexts and proved its security in the standard model from simple and non-interactive assumptions generically secure. We leave to future work the construction of fully secure (H)IBE systems in the symmetric key setting like defined by [SSW09]. 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.

References

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany.

[BDOP04] 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.

[BF03] Dan Boneh and Matthew K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[BGN05] 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.

[Boy03] 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.

[BW06] 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.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany.

[CHKP10] 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.

[Coc01] 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.

[GH09] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 437–456, San Francisco, CA, USA, 2009. Springer-Verlag, Berlin, Germany.

[HL02] 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.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate Encryption Supporting Disjunction, Polynomial Equations, and Inner Products. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162, Istanbul, Turkey, April 13–17, 2008. Springer-Verlag, Berlin, Germany.

[LOS+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.

[LW10] 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.

[Sha85] 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.

[SKOS09] 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.

[SSW09] 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.

[SW08] 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.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag, Berlin, Germany.

[Wat09] 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.