Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions

Brent Waters[⋆]

University of Texas at Austin bwaters@cs.utexas.edu

Abstract. We present a new methodology for proving security of encryption systems using what we call *Dual System Encryption*. Our techniques result in *fully* secure Identity-Based Encryption (IBE) and Hierarchical Identity-Based Encryption (HIBE) systems under the simple and established decisional Bilinear Diffie-Hellman and decisional Linear assumptions. Our IBE system has ciphertexts, private keys, and public parameters each consisting of a constant number of group elements. These results are the first HIBE system and the first IBE system with short parameters under simple assumptions.

In a Dual System Encryption system both ciphertexts and private keys can take on one of two indistinguishable forms. A private key or ciphertext will be *normal* if they are generated respectively from the system's key generation or encryption algorithm. These keys and ciphertexts will behave as one expects in an IBE system. In addition, we define *semi-functional* keys and ciphertexts. A semi-functional private key will be able to decrypt all normally generated ciphertexts; however, decryption will fail if one attempts to decrypt a semi-functional ciphertext with a semi-functional private key. Analogously, semi-functional ciphertexts will be decryptable only by normal private keys.

Dual System Encryption opens up a new way to prove security of IBE and related encryption systems. We define a sequence of games where we change first the challenge ciphertext and then the private keys one by one to be semi-functional. We finally end up in a game where the challenge ciphertext and all private keys are semi-functional at which point proving security is straightforward.

1 Introduction

The concept of Identity-Based Encryption (IBE) was first proposed by Shamir in 1984. In an IBE system a user can encrypt to another party simply by knowing their identity as well as a set of global parameters — eliminating the need to distribute a separate public key for each user in the system.

[⋆] Supported by NSF CNS-0716199, Air Force Office of Scientific Research (AFOSR) under the MURI award for "Collaborative policies and assured information sharing" (Project PRESIDIO) and the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001.

Although the concept received much interest, it wasn't until several years later that Boneh and Franklin [5] introduced the first Identity-Based Encryption scheme using groups with efficiently computable bilinear maps. The original Boneh and Franklin result used the random oracle heuristic to prove security under the Bilinear Diffie-Hellman assumption and a significant open question was whether the random oracle model could be removed.

Following the breakthrough result of Boneh and Franklin, there has been significant progress in realizing IBE in the standard model. First, Canetti, Halevi, and Katz [11] proved security without the random oracle heuristic, but under a weaker "Selective-ID" model where the attacker must declare the identity $\mathcal{I}^*$ that he will attack *before* even seeing the system's public parameters. Boneh and Boyen [2] then provided an efficient selectively secure scheme. Subsequently, Boneh and Boyen [3] and Waters [26] gave fully secure solutions in the standard model. The Waters scheme provided an efficient and provably fully secure system in the standard model under the decisional Bilinear Diffie-Hellman assumption; however, one drawback was that the public parameters consisted of $\mathcal{O}(\lambda)$ group elements for security parameter λ.

Partitioning Reductions One very important common thread in all of the above systems is that they use what we call a *partitioning* strategy to prove security. In these systems, one proves security to an underlying complexity assumption by creating a reduction algorithm $\mathcal{B}$ that partitions the identity space into two parts — 1) identities for which it can create private keys; and 2) identities that it can use in the challenge ciphertext phase. This partitioning is embedded either in the public parameters at setup time in the standard model systems [11, 2, 3, 26] or programed into the random oracle [5]. In the selective model, systems the identity space can be "tightly" partitioned so that all the keys except $\mathcal{I}^*$ fall into the key creating partition, while reductions in fully secure systems will partition the space according to the number of private key queries $q(\lambda)$ that an attacker makes and the reduction "hopes" that the queries and challenge ciphertext identity fall favorably in the partition.

While the partitioning techniques have proved useful, they have two fundamental limitations. First, the most efficient fully secure and standard model IBE system due to Waters has large public parameters that might be impractical for some applications. The second and more compelling concern is that partitioning techniques appear to be inadequate for proving security of encryption systems that offer more functionality such as Hierarchical IBE [20, 18] and Attribute-Based Encryption [22] *even if we apply the random oracle model.* For instance, all known Hierarchical Identity-Based Encryption (HIBE) systems (in this vein) have an exponential degradation of security with the depth, n, of the hierarchy — rendering the security reductions meaningless for large n. The fundamental problem is that more advanced systems such as HIBE have more structure on the identity space that make (any known) partitioning strategies unusable. For example, in an HIBE system a partitioning reduction algorithm is constrained such that if it can create a private key for a particular identity vector then it must be able to for all of its descendants.

Moving beyond the partitioning paradigm To overcome these obstacles, Gentry [15] proposed an IBE system with short public parameters that has a security reduction which moves beyond the partitioning paradigm. In his reduction the simulator is able to create a key for all identities and also use any identity as the challenge identity $\mathcal{I}^*$. At first glance, there is an apparent paradox in this strategy since it seems that the reduction algorithm could simply answer the challenge ciphertext itself by creating a private key for $\mathcal{I}^*$. To deal with this obstacle, Gentry's reduction algorithm can only generate *one* private key for each identity. For an attacker that makes at most q queries, the algorithm embeds a degree q polynomial $F(\cdot)$ and can create a private key with a tag component $F(\mathcal{I})$ for identity $\mathcal{I}$. The challenge ciphertext for $\mathcal{I}^*$ is structured such that it decrypts to the challenge message for the single key for $\mathcal{I}^*$ that the reduction could generate even though the message might be information theoretically hidden to an attacker with no knowledge of $F(\mathcal{I}^*)$.

Although the Gentry IBE achieved security in the standard model, it did so at the cost of using a significantly more complicated assumption called the decisional q-ABHDE assumption. In this assumption a generator g raised to several powers of an exponent a are given out (e.g., $g, g^a, g^{a^2}, \ldots, g^{a^q}$). In addition to the added complexity, the actual assumption used in the proof is *dependent* on the number of private key queries the adversary makes. This seems to be inherently tied to the need to embed the degree q polynomial f into a constant number group elements.

Interestingly, Gentry and Halevi [16] recently showed how to extend these concepts to get a fully secure HIBE system, although this system actually used an even more involved assumption. In addition, the "jump" from Gentry's IBE to the HIBE system added a significant amount of complexity to the system and proof of security.

Our Contribution We present a new methodology for proving security of encryption systems using what we call *Dual System Encryption*. Our techniques result in *fully* secure IBE and HIBE systems under the simple and established decisional Bilinear Diffie-Hellman and decisional Linear assumptions. Our IBE system has ciphertexts, private keys, and public parameters each consisting of a constant number of group elements. Our results give the first HIBE system and the first IBE system with short parameters under simple assumptions.

Our conceptual approach departs significantly from both the partitioning paradigm and Gentry's approach. In a Dual System Encryption system, both ciphertexts and private keys can take on one of two indistinguishable forms. A private key or ciphertext will be *normal* if they are generated respectively from the system's key generation or encryption algorithm. These keys and ciphertexts will behave as one expects in an IBE system. In addition, we define *semi-functional* keys and ciphertexts. A semi-functional private key will be able to decrypt all normally generated ciphertexts; however, decryption will fail if one attempts to decrypt a semi-functional ciphertext with a semi-functional private key. Analogously, semi-functional ciphertexts will be decryptable only by normal private keys.

Dual System Encryption opens up a new way to prove security of IBE and related encryption systems. Intuitively, to prove security we define a sequence of games arguing that an attacker cannot distinguish one game from the next. The first game will be the real security game in which the challenge ciphertext and all private keys are distributed normally. Next, we switch our normal challenge ciphertext with a semi- functional one. We argue that no adversary can detect this (under our complexity assumption) since all private keys given can decrypt the challenge ciphertext regardless of whether it is normal or semi-functional. In the next series of games, we change the private keys one game at a time from normal to semi-functional, again arguing indistinguishability. In both the above proof arguments, our reduction algorithm $\mathcal{B}$ will be able to provide private keys for any identity and use any identity as a challenge identity — eliminating the need to worry about an abort condition. Finally, we end up in a game where the challenge ciphertext and all private keys are semi-functional. At this point proving security is straightforward since the reduction algorithm does not need to present any normal keys to the attacker and all semi-functional keys are useless for decrypting a semi-functional ciphertext.

The reader may have noticed one issue in our indistinguishability argument over private keys. If the reduction algorithm $\mathcal{B}$ wants to know whether a secret key $\mathrm{SK}_{\mathcal{I}}$ for $\mathcal{I}$ was semi-functional, couldn't it simply create a semi-functional ciphertext for $\mathcal{I}$ and test this itself (without using the attacker)? To deal with this issue our reduction algorithm embeds a degree one polynomial $F(\mathcal{I}) = A \cdot \mathcal{I} + B$ (over $\mathbb{Z}_p$). In each hybrid game the attacker can only create a semi-functional ciphertext for ciphertext identity $\mathcal{I}_c$ with a "tag" value of $\mathrm{tag}_c = F(\mathcal{I}_c)$ and can only create a private key of unknown type for identity $\mathcal{I}_k$ with tag value of $\mathrm{tag}_k = F(\mathcal{I}_k)$. Our system use the "two equation revocation" technique of Sahai and Waters [23] to enforce that the decryption algorithm will only work if the key tag and ciphertext tag are not equal. If the reduction algorithm attempted to test the key in question, decryption would fail unconditionally; and thus independently of whether it was a semi-functional key. [1].

In reflection, one reason our dual system achieves security from a simple assumption is that by changing the keys in small hybrid steps one by one we only need to worry about the relationship between the challenge ciphertext and one private key at a time. Our function F only needs to be able to embed a degree one polynomial; in contrast the Gentry reduction "takes on" all private keys at the same time and needs a complex assumption to embed a degree q polynomial.

HIBE and Other Encryption Systems Building on our IBE system, we also provide a fully secure HIBE system. One remarkable feature is that the added complexity of the solution is rather small. Furthermore, our system combines the structure of the Boneh-Boyen [2] selective-ID HIBE. This hints that we can leverage our methodology to adapt ideas from other selectively secure encryption

[1] Our core system has a negligible correctness error; however, we outline how to build a perfectly correct system in Section 4.

systems (or those with complex assumptions) into fully secure ones under simple assumptions and also that prior selectively secure systems may have "lead us down the right path".

We believe that our Dual System methodology in the future will become a catalyst for proving adaptive security under simple assumptions for several other encryption systems including: Anonymous IBE and searchable encryption [4, 1, 10, 9, 24], Broadcast Encryption [14, 7], and Attribute-Based Encryption [22]. To add credence to this belief we give an adaptively secure broadcast system in the full version of our paper. Our broadcast system has ciphertext overhead of a constant number of group elements and is the first such system with a proof under a simple assumption.

Other Related Work We note that there are remarkable IBE systems of Cocks [13] and Boneh, Gentry, and Hamburg [6] based on the quadratic residuosity assumption and Gentry, Peikert, and Vaikuntanathan [17] based on lattice assumptions. These systems are all proven secure under the random oracle heuristic.

Katz and Wang [21] gave an IBE system with a tight security reduction in the random oracle model using a two-key approach. One might view this as falling outside the partition approach, although their techniques do not appear to give a path to full security for HIBE and related problems.

2 Background

We present a few facts related to groups with efficiently computable bilinear maps and then define the decisional Billinear-Diffie-Hellman and decisional Linear Assumptions. For space considerations, the definitions of security for Identity-Based Encryption and Hierarchical Identity-Based Encryption are included in our full version.

2.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and e be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.2 Decisional Bilinear Diffie-Hellman Assumption

We define the decisional Bilinear Diffie-Hellman problem as follows. Choose a group $\mathbb{G}$ of prime order p, where the size of p is a function of the security parameters. Next, choose a random generator g and random exponents $c_1, c_2, c_3 \in \mathbb{Z}_p$.

If an adversary is given

$$\boldsymbol{y} = g, g^{c_1}, g^{c_2}, g^{c_3},$$

it must remain hard to distinguish $e(g,g)^{c_1 c_2 c_3} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$.

An algorithm $\mathcal{B}$ that outputs $z \in \{0,1\}$ has advantage ϵ in solving decisional BDH problem in $\mathbb{G}$ if

$$\left| \Pr\left[\mathcal{B}\big(\boldsymbol{y}, T = e(g,g)^{c_1 c_2 c_3}\big) = 0\right] - \Pr\left[\mathcal{B}\big(\boldsymbol{y}, T = R\big) = 0\right] \right| \geq \epsilon \ .$$

Definition 1. *We say that the decisional BDH assumption holds if no polytime algorithm has a non-negligible advantage in solving the decisional BDH problem.*

2.3 Decisional Linear Assumption

We define the decisional Linear problem as follows. Choose a group $\mathbb{G}$ of prime order p, where the size of p is a function of the security paramters. Next, choose random generators g, f, ν and random exponents $c_1, c_2 \in \mathbb{Z}_p$. If an adversary is given

$$\boldsymbol{y} = g, f, \nu, g^{c_1}, f^{c_2},$$

it must remain hard to distinguish $\nu^{c_1 + c_2} \in \mathbb{G}$ from a random element in $\mathbb{G}$.

An algorithm $\mathcal{B}$ that outputs $z \in \{0,1\}$ has advantage ϵ in solving decisional Linear problem in $\mathbb{G}$ if

$$\left| \Pr\left[\mathcal{B}\big(\boldsymbol{y}, T = \nu^{c_1 + c_2}\big) = 0\right] - \Pr\left[\mathcal{B}\big(\boldsymbol{y}, T = R\big) = 0\right] \right| \geq \epsilon \ .$$

Definition 2. *We say that the decisional Linear assumption holds if no polytime algorithm has a non-negligible advantage in solving the decisional Linear problem.*

3 Identity-Based Encryption

We now present our core Identity-Based Encryption construction along with our proof of its security under the the decisional Linear and decisional BDH assumptions.

We first give the four algorithms of our IBE system. Next, we describe two additional algorithms for the creation of semi-functional ciphertexts and private keys respectively. The purpose of these algorithms is to define the structure of semi-functional ciphertexts and keys for our proof of security. We emphasize that these algorithms are not used in the actual system; indeed it is crucial for our security argument that no attacker could create ciphertexts or keys of this form.

Finally, we give the proof of our system against an attacker that makes at most q private key queries[2]. We organize our proof as a sequence of games. In the

[2] The maximum number of queries an attacker makes is, of course, a polynomial function $q(\cdot)$ of the security parameter; however, for notational simplicity we simply will speak of it making q private key queries.

sequence, we will gradually change the actual security game; first by introducing a semi-functional challenge ciphertext and then introduce semi-functional private keys one by one. We show that under the decisional Linear Assumption no adversary can distinguish between each successive game. Finally, we end up in a game where the challenge ciphertext and the *all* the private keys given out are semi-functional. At this point we can prove security under decisional-BDH.

3.1 Construction

Setup(λ) The authority first chooses a group $\mathbb{G}$ of prime order p. Next, it chooses generators $g, v, v_1, v_2, w, u, h \in \mathbb{G}$ and exponents $a_1, a_2, b, \alpha \in \mathbb{Z}_p$. Let $\tau_1 = vv_1^{a_1}, \tau_2 = vv_2^{a_2}$. It publishes the public parameters PK as the group description $\mathbb{G}$ along with:

$$g^b,\ g^{a_1},\ g^{a_2}, g^{b\cdot a_1},\ g^{b\cdot a_2},\ \tau_1, \tau_2, \tau_1^b, \tau_2^b,\ w,\ u,\ h, e(g,g)^{\alpha\cdot a_1\cdot b}.$$

The master secret key MSK consists of $g, g^\alpha, g^{\alpha\cdot a_1}, v, v_1, v_2$ as well as the public parameters. The identity space for the described scheme will be $\mathbb{Z}_p$, although we note in practice one can apply a collision resistant function to identities of arbitrary lengths.

Encrypt($\mathrm{PK}, \mathcal{I}, M$*)* The encryption algorithm chooses random s_1, s_2, t, and $\mathrm{tag}_c \in \mathbb{Z}_p$. Let $s = s_1 + s_2$. It then blinds $M \in \mathbb{G}_T$ as $C_0 = M \cdot (e(g,g)^{\alpha a_1 \cdot b})^{s_2}$ and creates:

$$C_1 = (g^b)^{s_1+s_2},\ C_2 = (g^{b\cdot a_1})^{s_1},\quad C_3 = (g^{a_1})^{s_1},\ C_4 = (g^{b\cdot a_2})^{s_2},\ C_5 = (g^{a_2})^{s_2},$$
$$C_6 = \tau_1^{s_1}\tau_2^{s_2},\ C_7 = (\tau_1^b)^{s_1}(\tau_2^b)^{s_2}w^{-t}, E_1 = (u^{\mathcal{I}}w^{\mathrm{tag}_c}h)^t,\ E_2 = g^t.$$

The ciphertext is $\mathrm{CT} = C_0, \ldots, C_7, E_1, E_2, \mathrm{tag}_c$.

KeyGen($\mathrm{MSK}, \mathcal{I}$*)* The authority chooses random $r_1, r_2, z_1, z_2, \mathrm{tag}_k \in \mathbb{Z}_p$. Let $r = r_1 + r_2$.

Then it creates:

$$D_1 = g^{\alpha\cdot a_1}v^r.\quad D_2 = g^{-\alpha}v_1^r g^{z_1}.\quad D_3 = (g^b)^{-z_1}.\quad D_4 = v_2^r g^{z_2},\quad D_5 = (g^b)^{-z_2}$$
$$D_6 = g^{r_2\cdot b},\quad D_7 = g^{r_1},\quad K = (u^{\mathcal{I}}w^{\mathrm{tag}_k}h)^{r_1}.$$

The secret key is $\mathrm{SK} = D_1, \ldots, D_7, K, \mathrm{tag}_k$.

Decrypt($\mathrm{CT}, K_{\mathcal{I}}$*)* The decryption algorithm will be able to decrypt a ciphertext encrypted for $\mathcal{I}$ with private key $\mathrm{SK}_{\mathcal{I}}$ if the ciphertext tag_c is not equal to the private key tag_k. Since both tags are chosen randomly, decryption will succeed with all but a negligible $1/p$ probability.

We break the decryption algorithm into a set of calculations. First, it computes:

$$\begin{aligned} A_1 &= e(C_1, D_1) \cdot e(C_2, D_2) \cdot e(C_3, D_3) \cdot e(C_4, D_4) \cdot e(C_5, D_5) \\ &= e(g,g)^{\alpha\cdot a_1\cdot b\cdot s_2} \cdot e(v,g)^{b(s_1+s_2)r} e(v_1,g)^{a_1 b s_1 r} e(v_2,g)^{a_2 b s_2 r}. \end{aligned}$$

Recall that $r = r_1 + r_2$. Next, it computes

$$\begin{aligned} A_2 &= e(C_6, D_6) \cdot e(C_7, D_7) \\ &= e(v, g)^{b(s_1+s_2)r} e(v_1, g)^{a_1 b s_1 r} e(v_2, g)^{a_2 b s_2 r} \cdot e(g, w)^{-r_1 t}. \end{aligned}$$

Taking, $A_3 = A_1/A_2 = e(g,g)^{\alpha \cdot a_1 \cdot b \cdot s_2} \cdot e(g,w)^{r_1 \cdot t}$ leaves us with one more cancellation to get the message blinding factor. If $\mathrm{tag}_c \neq \mathrm{tag}_k$ then the decryption algorithm can compute

$$A_4 = \big(e(E_1, D_7)/e(E_2, K)\big)^{1/(\mathrm{tag}_c - \mathrm{tag}_k)} = e(g,w)^{r_1 \cdot t}.$$

Finally, we can recover the message by computing

$$C_0/(A_3/A_4) = M.$$

Altogether, decryption requires nine applications of the pairing algorithm.

3.2 Semi-Functional Algorithms

We now describe the semi-functional ciphertext and key generation algorithms. We will define them as algorithms that are executed with knowledge of the secret exponents; however, in a real system they will not be used. Their main purpose is to define the structures that will be used in our proof. We define both semi-functional ciphertexts and keys in terms of a transformation on a normal ciphertext or key.

Semi-Functional Ciphertexts The algorithm first runs the encryption algorithm to generate a normal ciphertext CT for identity $\mathcal{I}$ and message M with $C'_1, \ldots, C'_7$,E'_1, E'_2. Then it chooses a random $x \in \mathbb{Z}_p$. It sets $C_1 = C'_1, C_2 = C'_2, C_3 = C'_3, E_1 = E'_1$,$E_2 = E'_2$, leaving these elements and the tag_c unchanged. It then sets

$$C_4 = C'_4 \cdot g^{ba_2x}, \quad C_5 = C'_5 \cdot g^{a_2x}, \quad C_6 = C'_6 \cdot v_2^{a_2x}, \quad C_7 = C'_7 \cdot v_2^{a_2bx}.$$

The semi-functional ciphertext is $C_1, \ldots, C_7, E_1, E_2, \mathrm{tag}_c$.

Semi-Functional Secret Keys The algorithm first runs the encryption algorithm to generate a normal private key $\mathrm{SK}_{\mathcal{I}}$ for identity $\mathcal{I}$ with $D'_1, \ldots, D'_7, K$. Then it chooses a random $\gamma \in \mathbb{Z}_p$. It sets $D_3 = D'_3, D_5 = D'_5, D_6 = D'_6, D_7 = D'_7, K = K'$, leaving these elements and the tag_k unchanged. It then sets

$$D_1 = D'_1 g^{-a_1 a_2 \gamma}, \quad D_2 = D'_2 \cdot g^{a_2 \gamma}, \quad D_4 = D'_4 \cdot g^{a_1 \gamma}$$

The semi-functional secret key is $\mathrm{SK} = D_1, \ldots, D_7, K, \mathrm{tag}_k$

Intuition We make a few remarks about the nature of the semi-functional keys and the structure of the system. First, we note that if one attempted to decrypt a semi-functional ciphertext with a normal key, then the decryption would succeed. This follows from the fact that

$$e(g^{ba_2 x}, D_4)e(g^{a_2 x}, D_5)/\big(e(v_2^{a_2 x}, D_6)e(v_2^{a_2 bx}, D_7)\big) = 1$$

when D_4, D_5, D_6, D_7 come from a normally generated ciphertext. One can view this as the extra "random" space occupied by the semi-functional part of the ciphertext as being orthogonal to the space defined by a normal key. For similar reasons, the semi-functional components of a private key will not impede decryption when applied on a normal ciphertext. However, when a semi-functional key is used to decrypt a semi-functional ciphertext decryption will fail (or end up giving a random message) because an extra $e(g, g)^{-a_1 a_2 x\gamma b}$ will be multiplied by the intended message.

We note that in order to generate semi-functional ciphertexts and private keys (according to the defined procedures) one respectively needs $v_2^{a_2 b}$ and $g^{a_1 a_2}$ — neither of which is available from the public parameters.

3.3 Proof of Security

We organize our proof as a sequence of games. The first game defined will be the real identity-based encryption game and the last one will be one in which the adversary has no advantage unconditionally. We will show that each game is indistinguishable from the next (under a complexity assumption). As stated before, the crux of our strategy is to move to a security game where both the challenge ciphertext and private keys are semi-functional. At this point any keys the challenger gives out are not useful in decrypting the ciphertext. We first define the games as:

Game$_{\textbf{Real}}$: The actual IBE security game defined in our full version.

Game$_i$: The real security game with the following two exceptions: 1) The challenge ciphertext will be a semi-functional ciphertext on the challenge identity $\mathcal{I}^*$. 2) The first i private key queries will return semi-functional private keys. The rest of the keys will be normal.

For an adversary that makes at most q queries we will be interested in **Game**$_0, \ldots,$ **Game**$_q$. We note that in **Game**$_0$ the challenge ciphertext is semi-functional, but all keys are normal and in **Game**$_q$ all private keys are semi-functional.

Game$_{\textbf{Final}}$: The real security game with the following exceptions: 1) The challenge ciphertext is a semi-functional encryption on a *random* group element of $\mathbb{G}_T$. 2) *All* of the private key queries result in semi-functional keys.

We now prove a set of Lemmas that argue about the distinguishablity of these games. For each proof we need to build a reduction simulator that both answers private key queries and creates a challenge ciphertext. We let **Game**$_{\text{Real}}\,\mathsf{Adv}_{\mathcal{A}}$ denote an algorithm $\mathcal{A}$'s advantage in the real game.

Lemma 1. *Suppose that there exists an algorithm $\mathcal{A}$ where* $\mathbf{Game}_{Real}\,\mathsf{Adv}_{\mathcal{A}} - \mathbf{Game}_{0}\,\mathsf{Adv}_{\mathcal{A}} = \epsilon$*. Then we can build an algorithm $\mathcal{B}$ that has advantage ϵ in the decision Linear game.*

Proof. Our algorithm $\mathcal{B}$ begins by taking in an instance $(\mathbb{G}, g, f, \nu, g^{c_1}, f^{c_2}, T)$ of the decision Linear problem. We now describe how it executes the Setup, Key Phase, and Challenge phases of the IBE game with $\mathcal{A}$.

Setup The algorithm chooses random exponents $b, \alpha, y_v, y_{v_1}, y_{v_2} \in \mathbb{Z}_p$ and random group elements $u, w, h \in \mathbb{G}$. It then sets $g = g, g^{a_1} = f, g^{a_2} = \nu$; intuitively a_1, a_2 are the exponents that the reduction cannot know itself.

Finally, it sets the variables as:

$$g^b,\ g^{b\cdot a_1} = f^b\ g^{b\cdot a_2} = \nu^b, v = g^{y_v}, v_1 = g^{y_{v_1}}, v_2 = g^{y_{v_2}}.$$

Using this it can calculate $\tau_1, \tau_2, \tau_1^b, \tau_2^b$ and $e(g,g)^{\alpha a_1 b} = e(g,f)^{\alpha\cdot b}$ in order to publish the public parameters PK. We also note that using α it can compute the master secret key for itself.

Key Generation Phases 1,2 Since $\mathcal{B}$ has the actual master secret key MSK it simply runs the key generation to generate the keys in both phases. Note that the MSK it has only allows for the creation of normal keys.

Challenge ciphertext $\mathcal{B}$ receives two messages M_0, M_1 and challenge identity $\mathcal{I}^*$. It then flips a coin β. We describe the creation of the challenge ciphertext in two steps. First, it creates a normal ciphertext using the real algorithm by calling Encrypt$(\text{PK}, \mathcal{I}^*, M_\beta)$, which outputs a ciphertext $\text{CT} = C'_0, \ldots, C'_7, E'_1, E'_2, \text{tag}_c$. Let s'_1, s'_2, t' be the random exponents used in creating the ciphertext.

Then we modify components of our ciphertext as follows. It sets

$$C_0 = C'_0 \cdot \big(e(g^{c_1}, f)\cdot e(g, f^{c_2})\big)^{b\cdot\alpha}, \quad C_1 = C'_1 \cdot (g^{c_1})^b, \quad C_2 = C'_2 \cdot (f^{c_2})^{-b},$$
$$C_3 = C'_3 \cdot (f^{c_2}), \quad C_4 = C'_4 \cdot (T)^b, C_5 = C'_5 \cdot T,$$
$$C_6 = C'_6 \cdot (g^{c_1})^{y_v} \cdot (f^{c_2})^{-y_{v_1}} \cdot T^{y_{v_2}}, \quad C_7 = C'_7 \cdot \big((g^{c_1})^{y_v} \cdot (f^{c_2})^{-y_{v_1}} \cdot T^{y_{v_2}}\big)^b,$$
$$E_1 = E'_1, \quad E_2 = E'_2.$$

The returned ciphertext is $\text{CT} = C_0, \ldots, C_7, E_1, E_2, \text{tag}_c$.

If T is a tuple, then this assignment implicitly sets $s_1 = -c_2 + s'_1, s_2 = s'_2 + c_1 + c_2$, and $s = s_1 + s_2 = c_1 + s'_1 + s'_2$. If $T = \nu^{c_1+c_2}$ it will have the same distribution as a standard ciphertext; otherwise, it will be distributed identically to a semi-functional ciphertext. $\mathcal{B}$ receives a bit β' and outputs 0 iff $\beta = \beta'$.

Lemma 2. *Suppose that there exists an algorithm $\mathcal{A}$ that makes at most q queries and* $\mathbf{Game}_{\mathsf{k-1}}\,\mathsf{Adv}_{\mathcal{A}} - \mathbf{Game}_{\mathsf{k}}\,\mathsf{Adv}_{\mathcal{A}} = \epsilon$ *for some k where $1 \le k \le q$. Then we can build an algorithm $\mathcal{B}$ that has advantage ϵ in the decision Linear game.*

Proof. Our algorithm $\mathcal{B}$ begins by taking in an instance $(\mathbb{G}, g, f, \nu, g^{c_1}, f^{c_2}, T)$ of the decision Linear problem. We now describe how it executes the Setup, Key Phase, and Challenge phases of the IBE game with $\mathcal{A}$.

Setup Algorithm $\mathcal{B}$ first chooses random exponents $\alpha, a_1, a_2, y_{v_1}, y_{v_2}, y_w, y_u, y_h$. It then defines

$$g = g, g^b = f, g^{b \cdot a_1} = f^{a_1},\ g^{b \cdot a_2} = f^{a_2},\ v = \nu^{-a_1 \cdot a_2},$$

$$v_1 = \nu^{a_2} \cdot g^{y_{v_1}},\ v_2 = \nu^{a_1} \cdot g^{y_{v_2}},\ e(g,g)^{\alpha \cdot a_1 b} = e(f,g)^{\alpha \cdot a_1}.$$

Now it can create

$$\tau_1 = v v_1^{a_1} = g^{y_{v_1} a_1} \quad \tau_2 = v v_1^{a_2} = g^{y_{v_2} a_2} \quad \tau_1^b = v v_1^{a_1} = f^{y_{v_1} a_1} \quad \tau_2^b = v v_1^{a_2} = f^{y_{v_2} a_2}.$$

Finally, $\mathcal{B}$ chooses random $A, B \in \mathbb{Z}_p$. It then sets

$$w = f g^{y_w}, \quad u = f^{-A} g^{y_u}, \quad h = f^{-B} g^{y_h}.$$

This will define all the public parameters of the system. Note that by virtue of knowing α, the algorithm $\mathcal{B}$ will know the regular master secret key.

We highlight the importance of the function $F(\mathcal{I}) = A \cdot \mathcal{I} + B$. One important feature is that for $\text{tag}_c = F(\mathcal{I})$ we have $(u^{\mathcal{I}} w^{\text{tag}_c} h) = f^{\text{tag}_c - A \cdot \mathcal{I} - B} g^{\mathcal{I} \cdot y_u + y_h + \text{tag}_c \cdot y_w}$ $= g^{\mathcal{I} \cdot y_u + y_h + \text{tag}_c \cdot y_w}$. In this case $\mathcal{B}$ will know the discrete log base g of the function. We also note that A, B are initially information theoretically hidden from the adversary. Since it is a pairwise independent function, if the adversary is given $F(\mathcal{I})$ for some identity, the, all values in $\mathbb{Z}_p$ are equally likely for $F(\mathcal{I}')$ for some $\mathcal{I} \neq \mathcal{I}'$.

Key Gen Phases 1,2 We break the Key Generation into three cases. Key Generation is done the same regardless of whether we are in phase 1 or 2.

Consider the i-th query made by $\mathcal{A}$.

Case 1: $i > k$

When i is greater than k our algorithm $\mathcal{B}$ will generate a normal key for the requested identity $\mathcal{I}$. Since it has the master secret key MSK it can run that algorithm.

Case 2: $i < k$

When i is less than k our algorithm $\mathcal{B}$ will generate a semi-functional key for the requested identity $\mathcal{I}$. It first creates a normal key using MSK. Then it makes it semi-functional using the procedure from above in Subsection 3.2. It can run this procedure since it knows $g^{a_1 a_2}$.

Case 3: $i = k$

The algorithm first runs the key generation algorithm to generate a normal private key $\text{SK}_{\mathcal{I}}$ for identity $\mathcal{I}$ with $D_1', \ldots, D_7', K$ using $\text{tag}_k{}^* = F(\mathcal{I})$. Let r_1', r_2', z_1', z_2' be the random exponents used.

It then sets

$$D_1 = D_1' \cdot T^{-a_1 \cdot a_2},\ D_2 = D_2' \cdot T^{a_2} (g^{c_1})^{y_{v_1}},\ D_3 = D_3' \cdot (f^{c_2})^{y_{v_1}},\ D_4 = D_4' \cdot T^{a_1} (g^{c_1})^{y_{v_2}},$$

$$D_5 = D_5' \cdot (f^{c_2})^{y_{v_2}},\ D_6 = D_6' \cdot f^{c_2},\ D_7 = D_7' \cdot (g^{c_1}),\ K = K' \cdot (g^{c_1})^{\mathcal{I} \cdot y_u + y_h + \text{tag}_k \cdot y_w}.$$

The semi-functional secret key is $\text{SK} = D_1, \ldots, D_7, K, \text{tag}_k$. We emphasize that the fact that $\text{tag}_k = F(\mathcal{I})$ allowed us to created the component K. In addition, we note that we implicitly set $z_1 = z_1' - y_{v_1} c_2$ and $z_2 = z_2' - y_{v_2} c_2$ in order to be able to create D_2 and D_4.

If T is a Linear tuple of the form $T = \nu^{c_1 + c_2}$, then the k-th query results in a normal key under randomness $r_1 = r_1' + c_1$ and $r_2 = r_2' + c_2$. Otherwise, if T is a random group element, then we can write $T = \nu^{c_1+c_2} g^{\gamma}$ for random $\gamma \in \mathbb{Z}_p$. This forms a semi-functional key where γ is the added randomness to make it semi-functional.

Challenge Ciphertext Algorithm $\mathcal{B}$ is given a challenge identity $\mathcal{I}^*$ and messages M_0, M_1. Then it flips a coin β.

In this phase $\mathcal{B}$ needs to be able to generate a semi-functional challenge ciphertext. One problem is that $\mathcal{B}$ does not have the group element v_2^b so it cannot directly create such a ciphertext. However, in the case where $\text{tag}_c{}^* = F(\mathcal{I}^*)$ it will have a different method of doing so.

$\mathcal{B}$ first runs the normal encryption algorithm to generate a normal ciphertext CT for identity $\mathcal{I}^*$ and message M^*; during this run it uses $\text{tag}_c{}^* = F(\mathcal{I}^*)$. It then gets a standard ciphertext $C_1', \ldots, C_7', E_1', E_2'$ under random exponents s_1', s_2', t'

To make it semi-functional it chooses a random $x \in \mathbb{Z}_p$. It first sets $C_1 = C_1', C_2 = C_2', C_3 = C_3'$ leaving these elements and the $\text{tag}_c{}^*$ unchanged. It then sets

$$C_4 = C_4'{\cdot}f^{a_2\cdot x}, \quad C_5 = C_5'{\cdot}g^{a_2\cdot x}, \quad C_6 = C_6'{\cdot}v_2^{a_2 x}, \quad C_7 = C_7'{\cdot}f^{y_{v_2}\cdot x\cdot a_2}\nu^{-a_1\cdot x\cdot y_w\cdot a_2},$$

$$E_1 = E_1' \cdot (\nu^{\mathcal{I}\cdot y_u + y_h + \text{tag}_c\cdot y_w})^{a_1 a_2 x} \quad E_2 = E_2' \cdot \nu^{a_1 a_2\cdot x}.$$

The semi-functional ciphertext is $C_1, \ldots, C_7, E_1, E_2, \text{tag}_c$.

Intuitively, the algorithm implicitly sets $g^t = g^{t'} + \nu^{a_1 a_2 x}$. This allows for the cancellation of the term $v_2^{a_1 a_2 b x}$ by w^{-t} in constructing C_7. Normally, this would be problematic for the generation of E_1; however since $\text{tag}_c{}^* = F(\mathcal{I}^*)$ $\mathcal{B}$ is able to create this term.

If T is a tuple, then we are in $\textbf{Game}_{k-1}$, otherwise we are in $\textbf{Game}_k$. We highlight that the adversary cannot detect any special relationship between $\text{tag}_c{}^*$ and $\text{tag}_k{}^*$ since $F(\mathcal{I}) = A \cdot \mathcal{I} + B$ is a pairwise independent function and A, B are hidden from its view.

$\mathcal{B}$ receives a bit β' and outputs 0 if $\beta = \beta'$.

Lemma 3. *Suppose that there exists an algorithm $\mathcal{A}$ that makes at most q queries and $\textbf{Game}_{\mathsf{q}}\,\mathsf{Adv}_{\mathcal{A}} - \textbf{Game}_{Final}\,\mathsf{Adv}_{\mathcal{A}} = \epsilon$. Then we can build an algorithm $\mathcal{B}$ that has advantage ϵ in the decision BDH game.*

Proof. We give the proof of security in the full version of our paper.

Theorem 1. *If the decisional Linear and decisional BDH assumptions hold then no poly-time algorithm can break our IBE system.*

Proof. Any attacker's advantage in $\mathbf{Game}_{\text{Final}}\,\mathsf{Adv}_{\mathcal{A}}$ in the final game must be 0 since it completely hides the bit β. By the sequence of games we established and Lemmas 1,2,3 an attacker's advantage in the real game $\mathbf{Game}_{\text{Real}}\,\mathsf{Adv}_{\mathcal{A}}$ must be negligibly close to 0.

4 Discussion

In this section we discuss a few potential future variations and implications of our IBE system.

Achieving Perfect Correctness Although having a negligible correctness error seems acceptable in practice, we would like to point out that we can close this gap by simply giving any user two private keys for an identity $\mathcal{I}$ each time they make a key request. The authority will simply run the original key generation algorithm twice with the restriction that the two key tags, $\text{tag}_{kA}, \text{tag}_{kB}$ are not equal. When attempting to decrypt a ciphertext at least one of the keys will work. The proof of security will work over each key piece — that is, each key request in the modified system will generate two distinct key requests (for the same identity) in the proof. We could also use a complementary two ciphertext approach and one private key approach.

Another potential solution is to run an efficient selectively secure IBE scheme [2] "in parallel". When a user encrypts a message M to $\mathcal{I}$ with tag_c in our original system, he will also encrypt M to the "identity" tag_c in the second selective system. A user with a key with tag_k will get a private key for "identity" tag_k in the second system. On decryption with $1 - 1/p$ probability the decryption algorithm will use the first ciphertext. However, if the tags align it can use the second ciphertext.

Signature Scheme Naor [3] observed that any (fully secure) IBE system gives rise to a signature scheme secure under the same assumptions. The signature system from our IBE scheme has the favorable properties that the public parameters and signatures are a constant number of group elements, it is provable in the standard model, and it is stateless. While some previous signature schemes derived from IBE systems (e.g. BLS [8] or Waters [26] signatures) depended on the computational variants of the assumptions, our proof technique seems to require the decisional Linear Assumption. One interesting approach would be to see if one could create shorter signatures than those generated in the generic conversion by using the IBE systems private keys.

Chosen Ciphertext Security We note that using the transformation of Canetti, Halevi, and Katz [12] we can achieve chosen ciphertext security from the HIBE scheme of Section 5.

[3] The observation was documented by Boneh and Franklin [5].

Security under the XDH Assumption One factor in the size and complexity of our IBE system is that it relies upon the Decisional Linear Assumption to hide the form of both keys and ciphertexts. One potential alternative is to use asymmetric bilinear groups, where we have $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Using these group we might assume DDH is hard both within $\mathbb{G}_1$ and within $\mathbb{G}_2$; this has also been called the XDH assumption. Using this assumption we might hope to shave off three group elements from both ciphertexts and private keys.

Alternative to Incompleteness An critical part to arguing security is that an attacker could not distinguish normal keys from semi-functional ones. Our approach was to use a hybrid argument where for the key in question its $\text{tag}_k = F(\mathcal{I})$. If the simulator attempted to create the key in question for $\mathcal{I}^*$ and test it on the challenge ciphertext this would not work since $\text{tag}_k = \text{tag}_c$. Intuitively, the simulator could not test whether the key was semi-functional since decryption would fail *regardless* of whether the key was semi-functional or not. One might consider taking the opposite approach where decryption would always succeed if $\text{tag}_c = \text{tag}_k$ even if both the key and ciphertext are semi-functional. We note this approach would also require a slightly different proof strategy for proving Lemma 3.

5 Hierarchical Identity-Based Encryption

In this section we present our Hierarchical Identity-Based Encryption system. Our construction will build on top of our IBE scheme of Section 3. The reader will notice that the added complexity of moving from our IBE to HIBE system is remarkably simple. The same core concepts of our construction and proof methodology apply. One might view the HIBE system as "combining" the structure of the Boneh-Boyen [2] HIBE system with our techniques to get full security.

One challenging aspect in the proof of security is that a private key of depth d will have associated tags: $\text{tag}_{k1}, \ldots, \text{tag}_{kd}$. If we run our delegation algorithm to create a new key of depth $d+1$, the new key will inherit the previous key's tag values and there is no method for "re-randomizing" them. Most prior security definitions of HIBE [20, 18] define a game where all keys come from an authority and don't model any distinctions on how a key was created (i.e. trace paths of delegation). The prior definitions are only valid if keys from the delegation algorithm are distributed identically to a fresh call to the key generation algorithm [4]; however, due to the "tag lineage" described this is clearly not the case. To argue security we use a "complete" model of HIBE security introduced by Shi and Waters [25] that we define in our full version. Due to space considerations our proof of security is also in the full version.

5.1 Construction

In our system we will consider a hierarchical identity as an identity vector $\boldsymbol{I} = \mathcal{I}_1 : \cdots : \mathcal{I}_d$ for some depth d, where $d \leq n$ for some maximum depth n. We

[4] This is actually the case for most prior systems, so the proofs of security do hold up.

assume that the identities are encoded such that for two identities $\boldsymbol{I}, \boldsymbol{I'}$ if $\mathcal{I}_i = \mathcal{I}'_i$ then $\mathcal{I}_j = \mathcal{I}'_j$ for all $j \leq i$. We can enforce this by encoding all previous levels. For example, an identity of level one "com" and level two "yahoo" can be encoded as "com":"com.yahoo", where '.' is a special symbol. In practice, one will use a collision resistant hash function to hash identities of arbitrary length to $\mathbb{Z}_p$.

Setup(λ, n*)* The setup algorithm takes as input a security parameter and the maximum depth n. The authority first chooses a group $\mathbb{G}$ of prime order p. Next, it chooses generators $g, v, v_1, v_2, w,\ u_1, \ldots, u_n, h_1, \ldots, h_n \in \mathbb{G}$ and exponents $a_1, a_2, b, \alpha \in \mathbb{Z}_p$. Let $\tau_1 = vv_1^{a_1}, \tau_2 = vv_2^{a_2}$. It publishes the public parameters PK as the group description $\mathbb{G}$ along with:

$$g^b,\ g^{a_1},\ g^{a_2}, g^{b \cdot a_1},\ g^{b \cdot a_2},\ \tau_1, \tau_2, \tau_1^b, \tau_2^b, v,\ v_1,\ v_2,\ \ w,\ u_1, \ldots, u_n,$$

$$h_1, \ldots, h_n, e(g,g)^{\alpha \cdot a_1 \cdot b}.$$

The master secret key MSK consists of $g, g^{\alpha}, g^{\alpha \cdot a_1}$ as well as the public parameters. The identity space for the described scheme will be $\mathbb{Z}_p$.

*Encrypt(*PK, $\boldsymbol{I} = \mathcal{I}_1 : \cdots : \mathcal{I}_d, M$*)* The encryption algorithm will encrypt to an identity vector of depth $d \leq n$. It chooses random $s_1, s_2, t \in \mathbb{Z}_p$ and $\mathrm{tag}_{c1}, \ldots, \mathrm{tag}_{cd} \in \mathbb{Z}_p$. Let $s = s_1 + s_2$. It then blinds $M \in \mathbb{G}_T$ as $C_0 = M \cdot (e(g,g)^{\alpha a_1 \cdot b})^{s_2}$ and creates:

$$C_1 = (g^b)^{s_1+s_2},\ C_2 = (g^{b \cdot a_1})^{s_1},\ C_3 = (g^{a_1})^{s_1},\ C_4 = (g^{b \cdot a_2})^{s_2},\ C_5 = (g^{a_2})^{s_2},$$

$$C_6 = \tau_1^{s_1} \tau_2^{s_2},\ C_7 = (\tau_1^b)^{s_1} (\tau_2^b)^{s_2} w^{-t},$$

$$E_1 = (u_1^{\mathcal{I}_1} w^{\mathrm{tag}_{c1}} h_1)^t, \ldots, E_d = (u_d^{\mathcal{I}_d} w^{\mathrm{tag}_{cd}} h_d)^t, \quad \tilde{E} = g^t.$$

The ciphertext is $\mathrm{CT} = C_0, \ldots, C_7, E_1, , \ldots, E_d, \tilde{E}, \mathrm{tag}_{c1}, \ldots, \mathrm{tag}_{kd}$.

*KeyGen(*MSK, $\boldsymbol{I} = \mathcal{I}_1 : \cdots : \mathcal{I}_d$*)* The authority chooses random $\mu_1, \ldots, \mu_d$ $, r_2, z_1, z_2, \mathrm{tag}_{k1}, \ldots, \mathrm{tag}_{kd} \in \mathbb{Z}_p$. First let $r_1 = \sum_{1 \leq i \leq d} \mu_i$ and then let $r = r_1 + r_2$. Then it creates:

$$D_1 = g^{\alpha \cdot a_1} v^r, \quad D_2 = g^{-\alpha} v_1^r g^{z_1}, \quad D_3 = (g^b)^{-z_1}, \quad D_4 = v_2^r g^{z_2}, \quad D_5 = (g^b)^{-z_2},$$

$$D_6 = g^{r_2 \cdot b}, \quad D_7 = g^{r_1},$$

$$(K_{1,1} = (u_1^{\mathcal{I}_1} w^{\mathrm{tag}_{k1}} h_1)^{\mu_1},\ K_{1,2} = g^{\mu_1}.), \ldots, (K_{d,1} = (u_d^{\mathcal{I}_d} w^{\mathrm{tag}_{kd}} h_d)^{\mu_d},\ K_{d,2} = g^{\mu_d})$$

The secret key is $\mathrm{SK} = D_1, \ldots, D_7, (K_{1,1}, K_{1,2}), \ldots, (K_{d,1}, K_{d,2}) \mathrm{tag}_{k1}, \ldots, \mathrm{tag}_{kd}$.

Delegate($\mathrm{PK}, \mathrm{SK}_{\boldsymbol{I}=\mathcal{I}_1:\cdots:\mathcal{I}_d}, \mathcal{I}_{d+1}$*)* The algorithm will take a secret key $\mathrm{SK} = D'_1, \ldots, D'_7$, $(K'_{1,1}, K'_{1,2})$, $\ldots$, $(K'_{d,1}, K'_{d,2})$, $\mathrm{tag}_{k1}, \ldots, \mathrm{tag}_{kd}$ for $\boldsymbol{I}$ and extend it to depth $d+1$ by creating a key for $\boldsymbol{I} : \mathcal{I}_{d+1}$.

The algorithm will "re-randomize" the existing key in the process of appending on a new key component; however, the existing tag_k values will remain. It chooses random $\mu_1, \ldots, \mu_{d+1}, r_2, z_1, z_2$,$\mathrm{tag}_{kd+1} \in \mathbb{Z}_p$. First let $r_1 = \sum_{1 \le i \le d+1} \mu_i$ and then let $r = r_1 + r_2$. Then it creates:

$$D_1 = D'_1 \cdot v^r, \quad D_2 = D'_2 \cdot v_1^r g^{z_1}, \quad D_3 = D'_3 \cdot (g^b)^{-z_1}, \quad D_4 = D'_4 \cdot v_2^r g^{z_2},$$

$$D_5 = D'_5 \cdot (g^b)^{-z_2}, \quad D_6 = D'_6 \cdot g^{r_2 \cdot b}, \quad D_7 = D'_7 \cdot g^{r_1},$$

$$K_{1,1} = K'_{1,1} \cdot (u_1^{\mathcal{I}_1} w^{\mathrm{tag}_{k1}} h_1)^{\mu_1}, \ldots, K_{d,1} = K'_{d,1} \cdot (u_d^{\mathcal{I}_d} w^{\mathrm{tag}_{kd}} h_d)^{\mu_d},$$

$$K_{d+1,1} = (u_{d+1}^{\mathcal{I}_{d+1}} w^{\mathrm{tag}_{kd+1}} h_{d+1})^{\mu_{d+1}},$$

$$K_{1,2} = K'_{1,2} \cdot g^{\mu_1}, \ldots, K_{d,2} = K'_{d,2} \cdot g^{\mu_d}, \quad K_{d+1,2} = g^{\mu_{d+1}}.$$

The secret key is $\mathrm{SK} = D_1, \ldots, D_7$,$(K_{1,1}, K_{1,2})$,$\ldots$,$(K_{d+1,1}, K_{d+1,2})$,$\mathrm{tag}_{k1}, \ldots$,tag_{kd+1}.

Decrypt($\mathrm{CT}, K_{\mathcal{I}}$*)* The decryption algorithm will be able to decrypt a ciphertext encrypted for $\boldsymbol{I}'$ of depth d' with private key $\mathrm{SK}_{\boldsymbol{I}}$ of depth d if 1) $\forall i \le d : \ \boldsymbol{I}'_i = \boldsymbol{I}_i$ for all $i \le d$ and 2) $\forall i \le d : \ \mathrm{tag}_{ci} \ne \mathrm{tag}_{ki}$. We break the decryption algorithm into a set of calculations: First, it computes:

$$A_1 = e(C_1, D_1) \cdot e(C_2, D_2) \cdot e(C_3, D_3) \cdot e(C_4, D_4) \cdot e(C_5, D_5)$$

$$A_2 = e(C_6, D_6) \cdot e(C_7, D_7) \quad A_3 = A_1/A_2 = e(g,g)^{\alpha \cdot a_1 \cdot b \cdot s_2} \cdot e(g,w)^{r_1 \cdot t}.$$

If $\forall i \le d$ we have $\mathrm{tag}_{ci} \ne \mathrm{tag}_{ki}$ then the decryption algorithm can compute

$$A_4 = \big(e(E_1, K_{1,2})/e(\tilde{E}, K_{1,1})\big)^{1/(\mathrm{tag}_{c1} - \mathrm{tag}_{k1})} \cdots$$

$$\big(e(E_d, K_{d,2})/e(\tilde{E}, K_{d,1})\big)^{1/(\mathrm{tag}_{cd} - \mathrm{tag}_{kd})} = e(g,w)^{t \sum_{1 \le d} \mu_i}.$$

Finally, we can recover the message by computing $C_0/(A_3/A_4) = M$.

References

[1] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In *CRYPTO*, pages 205–222, 2005.

[2] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[3] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pages 443–459, 2004.

[4] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[5] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.

[6] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *FOCS*, pages 647–657, 2007.

[7] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.

[8] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *ASIACRYPT*, pages 514–532, 2001.

[9] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007.

[10] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO*, pages 290–307, 2006.

[11] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

[12] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.

[13] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[14] Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO*, pages 480–491, 1993.

[15] Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[16] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In *TCC*, 2009.

[17] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

[18] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT*, pages 548–566, 2002.

[19] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems. In *Eurocrypt*, 2009.

[20] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT*, pages 466–481, 2002.

[21] Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. In *ACM Conference on Computer and Communications Security*, pages 155–164, 2003.

[22] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[23] Amit Sahai and Brent Waters. Revocation systems with very small private keys. Cryptology ePrint Archive, Report 2008/309, 2008.

[24] Elaine Shi, John Bethencourt, Hubert T.-H. Chan, Dawn Xiaodong Song, and Adrian Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy*, pages 350–364, 2007.

[25] Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In *ICALP (2)*, pages 560–578, 2008.

[26] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.