Full version of an extended abstract published in Advances in Cryptology—CRYPTO 2006, Springer-Verlag, 2006. Available from the IACR Cryptology ePrint Archive as Report 2006/085.

Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles)

Xavier Boyen [*] Brent Waters [†]

June 8, 2006

Abstract

We present an identity-based cryptosystem that features fully anonymous ciphertexts and hierarchical key delegation. We give a proof of security in the standard model, based on the mild Decision Linear complexity assumption in bilinear groups. The system is efficient and practical, with small ciphertexts of size linear in the depth of the hierarchy. Applications include search on encrypted data, fully private communication, *etc.*

Our results resolve two open problems pertaining to anonymous identity-based encryption, our scheme being the first to offer provable anonymity in the standard model, in addition to being the first to realize fully anonymous HIBE at all levels in the hierarchy.

1 Introduction

The cryptographic primitive of Identity-Based Encryption (IBE) allows a sender to encrypt a message for a receiver using only the receiver's identity as a public key. Recently, there has been interest in "anonymous" identity-based encryption systems, where the ciphertext does not leak the identity of the recipient. In addition to their obvious privacy benefits, anonymous IBE systems can be leveraged to construct Public key Encryption with Keyword Search (PEKS) schemes, as was first observed by Boneh *et al.* [10] and later formalized by Abdalla *et al.* [1]. Roughly speaking, PEKS is a form of public key encryption that allows an encryptor to make a document serarchable by keywords, and where the capabilities to search on particular keywords are delegated by a central authority. Anonymous HIBE further enables sophisticated access policies for PEKS and ID-based PEKS.

Prior to this paper, the only IBE system known to be inherently anonymous was that of Boneh and Franklin [11]. Although they did not state it explicitly, the anonymity of their scheme followed readily from their proof of semantic security. This was noticed by Boyen [13], who gave an ID-based signcryption with a formalization of sender and recipient anonymity. One drawback of the Boneh-Franklin IBE paradigm is that its security proofs are set in the random oracle model. More recently, a number of IBE schemes [15, 5, 6, 32, 17, 27] have been proven secure outside of the random oracle model, but none of these schemes is anonymous. In particular, in the efficient schemes of Boneh and Boyen [5] and Waters [32], the identity is deterministically encoded in a simple manner within

[*]Voltage Inc., Palo Alto — `xb@boyen.org`
[†]SRI International — `bwaters@csl.sri.com`

the exponent of an element of the bilinear group $\mathbb{G}$. When these schemes are implemented using a "symmetric" bilinear pairing $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, it becomes trivial to test whether a given ciphertext was encrypted for a candidate identity.

A tempting workaround to this problem is to use an "asymmetric" pairing $\mathbf{e} : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ in the schemes that allow it, such as Boneh and Boyen's "BB_1" and "BB_2", and Waters' by extension. In those schemes, and under the additional assumption that Decision Diffie-Hellman is hard in $\mathbb{G}$, one may prevent the use of the pairing as a direct test of whether a ciphertext is for a particular identity. Unfortunately, turning this observation into a formal security reduction would at the very least require making a strong assumption that is patently false in bilinear groups with symmetric pairings, and the approach would still fail to generalize to hierarchical IBE for fundamental reasons that are discussed later. Ideally, one would like a scheme that works indifferently with symmetric and asymmetric pairings, and generalizes to hierarchical identities.

The first anonymous IBE without random oracles was unveiled at the CRYPTO'05 Rump Session by one of the authors, and is now described in Section 4. In a nutshell, the identity is split randomly into two blind components to prevent its recognition by using the bilinear map, without making unduly strong assumptions. A second anonymous IBE without random oracles was recently proposed by Gentry [20], based on a different principle. In Gentry's scheme, the identity of a ciphertext cannot be tested because a crucial element of the ciphertext lives in the target group $\mathbb{G}_T$ rather than the bilinear group $\mathbb{G}$. Gentry's scheme is very efficient and has a number of advantages, but unfortunately relies on a strong complexity assumption and does not generalize to hierarchical IBE.

In spite of these recent achievements, creating an Anonymous Hierarchical IBE (A-HIBE) scheme has remained a challenge. Even if we avail ourselves of random oracles, there simply does not exist any known hierarchical identity-based encryption scheme which is also anonymous. In particular, the Gentry-Silverberg [21] HIBE scheme is not anonymous, despite the fact that it derives from the Boneh-Franklin IBE scheme, which is anonymous. The numerous applications to searching on encrypted data motivated Abdalla *et al.* [1], in their CRYPTO'05 paper, to ask for the creation of an Anonymous HIBE system, preferably without random oracles, as an important open research problem.

1.1 Our Results

Our contribution is twofold. First, we build a simple and efficient Anonymous IBE scheme, and give a proof of security without random oracles. Second, we generalize our construction into a fully Anonymous HIBE scheme (*i.e.*, anonymous at all levels in the hierarchy), again with a proof without random oracles. Our approach gives a very efficient system in the non-hierarchical case, and remains practical for the shallow hierarchies that are likely to be encountered in most applications. The security of our systems is based on Boneh's *et al.* [9] Decision Linear assumption, which is arguably one of the mildest useful complexity assumptions in the realm of bilinear groups.

At first sight, our construction bears a superficial resemblance to Boneh and Boyen's "BB_1" HIBE scheme [5, §4] — but with at least two big differences. First, we perform "linear splittings" on various portions of the ciphertext, to thwart the trial-and-error identity guessing to which other schemes fell prey. This idea gives us provable anonymity, even under symmetric pairings. Second, we use multiple parallel HIBE systems and re-randomize the keys between them upon each delegation. This is what lets us use the linear splitting technique at all levels of the hierarchy, but also poses a technical challenge in the security reduction which must now simulate multiple interacting HIBE systems *at once*. Solving this problem was the crucial step that gave us a hierarchy without destroying anonymity.

1.2 Related Work

The concept of identity-based encryption was first proposed by Shamir [29] two decades ago. However, it was not until much later that Boneh and Franklin [11] and Cocks [18] presented the first practical solutions. The Boneh-Franklin IBE scheme was based on groups with efficiently computable bilinear maps, while the Cocks scheme was proven secure under the quadratic residuosity problem, which relies on the hardness of factoring. The security of either scheme was only proven in the random oracle model.

Canetti, Halevi, and Katz [15] suggested a weaker security notion for IBE, known as selective identity or selective-ID, relative to which they were able to build an inefficient but secure IBE scheme without using random oracles. Boneh and Boyen [5] presented two very efficient IBE systems ("BB_1" and "BB_2") with selective-ID security proofs, also without random oracles. The same authors [6] then proposed a coding-theoretic extension to their "BB_1" scheme that allowed them to prove security for the full notion of adaptive identity or adaptive-ID security without random oracles, but the construction was impractical. Waters [32] then proposed a much simpler extension to "BB_1" also with an adaptive-ID security proof without random oracles; its efficiency was further improved in two recent independent papers, [17] and [27].

The notion of hierarchical identity-based encryption was first defined by Horwitz and Lynn [23], and a construction in the random oracle model given by Gentry and Silverberg [21]. Canetti, Halevi, and Katz [15] give the first HIBE with a (selective-ID) security proof without random oracles, but that is not efficient. The first efficient HIBE scheme to be provably secure without random oracles is the "BB_1" system of Boneh and Boyen; further improvements include the HIBE scheme by Boneh, Boyen, and Goh [7], which features shorter ciphertexts and private keys.

Nominally adaptive-ID secure HIBE schemes have been proposed, although all constructions known to date [21, 32, 17, 27] are depth-limited because they suffer from an exponential security degradation with the depth of the hierarchy. Qualitatively, this is no different than taking an HIBE scheme with tight selective-ID security, such as BB_1 or BBG, and using one of the generic transformations from [5, §7] to make it adaptive-ID secure. Quantitatively, the rate of decay will differ between those approaches, which means that the number of useful hierarchy levels will evolve similarly but not identically in function of the chosen group size and the desired security bit strength. Accordingly, it remains an important open problem in identity-based cryptography to devise an adaptive-ID secure HIBE scheme whose security degrades at most polynomially with the depth of the hierarchy, under reasonable assumptions. (In this paper, we mostly leave aside this issue of adaptive-ID security for HIBE.)

Encrypted search was studied by Song, Wagner, and Perrig [31], who presented the first scheme for searching on encrypted data. Their scheme is in the symmetric-key setting where the same party that encrypted the data would generate the keyword search capabilities. Boneh *et al.* [10] introduced Public Key Encryption with Keyword Search (PEKS), where any party with access to a public key could make an encrypted document that was searchable by keyword; they realized their construction by applying the Boneh-Franklin IBE scheme. Abdalla *et al.* [1] recently formalized the notion of Anonymous IBE and its relationship to PEKS. Additionally, they formalized the notion of Anonymous HIBE and mentioned different applications for it. Using the GS system as a starting point, they also gave an HIBE scheme that was anonymous at the first level, in the random oracle model. Another view of Anonymous IBE is as a combination of identity-based encryption with the property of key privacy, which was introduced by Bellare *et al.* [4].

1.3 Applications

In this section we discuss various applications of our fully anonymous HIBE system. The main applications can be split into several broad categories.

Fully Private Communication. The first compelling application of anonymous IBE is for fully private communication. Bellare *et al.* [4] argue that public key encryption systems that have the "key privacy" property can be used for anonymous communication: for example, if one wishes to hide the identity of a recipient one can encrypt a ciphertext with an anonymous IBE system and post it on a public bulletin board. By the anonymity property, the ciphertext will betray neither sender nor recipient identity, and since the bulletin board is public, this method will also be resistant to traffic analysis. To compound this notion of key privacy, identity-based encryption is particularly suited for untraceable anonymous communication, since, contrarily to public-key infrastructures, the sender does not even need to query a directory for the public key of the recipient. For this reason, anonymous IBE provides a very convincing solution to the problem of secure anonymous communication, as it makes it harder to conduct traffic analysis attack on directory lookups.

Search on Encrypted Data. The second main application of anonymous (H)IBE is for encrypted search. As mentioned earlier, anonymous IBE and HIBE give several application in the Public-key Encryption with Keyword Search (PEKS) domain, proposed by Boneh *et al.* [10], and further discussed by Abdalla *et al.* [1]. As a simple example of real-world application of our scheme, PEKS is a useful primitive for building secure audit logs [33, 19]. Furthermore, one can leverage the hierarchical identities in our anonymous HIBE in several interesting ways. For example, we can use a two-level anonymous HIBE scheme where the first level is an identity and the second level is a keyword. This gives us the first implementation of the Identity-Based Encryption with Keyword Search (IBEKS) primitive asked for in [1]. With this primitive, someone with the private key for an identity can delegate out search capabilities for encryptions to their identity, without requiring a central authority to act as the delegator. Conversely, by using certain keywords such as "Top Secret" at the first level of the hierarchy, it is possible to broadcast innocent-looking ciphertexts that require a certain clearance to decrypt, without even hinting at the fact that their payload might be valuable. We can create more refined search capabilities with a deeper hierarchy.

As the last applications we mention, forward-secure public-key encryption [15] and forward-secure HIBE [34] are not hard to construct from HIBE systems with certain algebraic properties [7]. Without going into details, we mention that we can implement Anonymous fs-HIBE with our scheme by embedding a time component within the hierarchy, while preserving the anonymity property.

2 Background

Recall that a pairing is an efficiently computable [26], non-degenerate function, $\mathbf{e} : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$, with the bilinearity property that $\mathbf{e}(g^r, \hat{g}^s) = \mathbf{e}(g, \hat{g})^{r\,s}$. Here, $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\mathbb{G}_T$ are all multiplicative groups of prime order p, respectively generated by g, $\hat{g}$, and $\mathbf{e}(g, \hat{g})$. It is *asymmetric* if $\mathbb{G} \neq \hat{\mathbb{G}}$.

We call *bilinear instance* a tuple $\mathbf{G} = [p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, g, \hat{g}, \mathbf{e}]$. We assume an efficient generation procedure that on input a security parameter $\Sigma \in \mathbb{N}$ outputs $\mathbf{G} \xleftarrow{\$} \mathrm{Gen}(1^\Sigma)$ where $\log_2(p) = \Theta(\Sigma)$. We write $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$ for the set of residues $\bmod p$ and $\mathbb{Z}_p^\times = \mathbb{Z}_p \setminus \{0\}$ for its multiplicative group.

2.1 Assumptions

Since bilinear groups first appeared in cryptography half a decade ago [24], several years after their first use in cryptanalysis [25], bilinear maps or pairings have been used in a large variety of ways under many different complexity assumptions. Some of them are very strong; others are weaker. Informally, we say that an assumption is *mild* if it is tautological in the generic group model [30], and also "efficiently falsifiable" [28] in the sense that its problem instances are stated non-interactively and concisely (*e.g.*, independently of the number of adversarial queries or such large quantity). Most IBE and HIBE schemes mentioned in Introduction (except "BB_2" and the Factoring-based system by Cocks) are based on *mild* bilinear complexity assumptions, such as BDH [24, 11] and Linear [9]. In this paper, our goal is to rely only on mild assumptions.

Decision BDH: The Bilinear DH assumption was first used by Joux [24], and gained popularity for its role in the Boneh-Franklin IBE system [11]. The decisional assumption posits the hardness of the D-BDH problem, which we state in asymmetric bilinear groups as:

> Given a tuple $[g, g^{z_1}, g^{z_3}, \hat{g}, \hat{g}^{z_1}, \hat{g}^{z_2}, Z] \in \mathbb{G}^3 \times \hat{\mathbb{G}}^3 \times \mathbb{G}_T$ for random exponents $[z_1, z_2, z_3] \in (\mathbb{Z}_p)^3$, decide whether $Z = \mathbf{e}(g, \hat{g})^{z_1 z_2 z_3}$.

Decision Linear: The Linear assumption was first proposed by Boneh, Boyen, and Shacham for group signatures [9]. Its decisional form posits the hardness of the D-Linear problem, which can be stated in asymmetric bilinear groups as follows:

> Given a tuple $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, \hat{g}, \hat{g}^{z_1}, \hat{g}^{z_2}, Z] \in \mathbb{G}^5 \times \hat{\mathbb{G}}^3 \times \mathbb{G}$ for random $[z_1, z_2, z_3, z_4] \in (\mathbb{Z}_p)^4$, decide whether $Z = g^{z_3 + z_4}$.

We remark that the elements $\hat{g}, \hat{g}^{z_1}, \hat{g}^{z_2} \in \hat{\mathbb{G}}^3$ were not explicitly included in Boneh's *et al.* original formulation.

"Hard" means algorithmically non-solvable with probability $1/2 + \Omega(\mathrm{poly}(\Sigma)^{-1})$ in time $\mathcal{O}(\mathrm{poly}(\Sigma))$ for efficiently generated random "bilinear instances" $[p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, g, \hat{g}, \mathbf{e}] \xleftarrow{\$} \mathrm{Gen}(1^\Sigma)$, as $\Sigma \to +\infty$.

These assumptions allow but not require the groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ to be distinct, and similarly we make no representation one way or the other regarding the existence of computable homomorphisms between $\mathbb{G}$ and $\hat{\mathbb{G}}$, in either direction. This is the most general formulation. It has two main benefits: (1) since it comes with fewer restrictions, it is potentially more robust and increases our confidence in the assumptions we make; and (2) it gives us the flexibility to implement the bilinear pairing on a broad variety of algebraic curves with attractive computational characteristics [2], whereas symmetric pairings tend to be confined to supersingular curves, to name this one distinction.

Note that if we let $\mathbb{G} = \hat{\mathbb{G}}$ and $g = \hat{g}$, our assumptions regain their familiar "symmetric" forms:

Given $[g, g^{z_1}, g^{z_2}, g^{z_3}, Z] \in \mathbb{G}^4 \times \mathbb{G}_T$ for random $[z_1, z_2, z_3] \in (\mathbb{Z}_p)^3$, decide whether $Z = \mathbf{e}(g, g)^{z_1 z_2 z_3}$.
Given $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, Z] \in \mathbb{G}^5 \times \mathbb{G}$ for random $[z_1, z_2, z_3, z_4] \in (\mathbb{Z}_p)^4$, decide if $Z = g^{z_3 + z_4}$.

As a rule of thumb, the remainder of this paper may be read in the context of symmetric pairings, simply by dropping all "hats" ($\hat{}$) in the notation. Also note that D-Linear trivially implies D-BDH.

2.2 Models

We briefly precise the security notions that are implied by the concept of Anonymous IBE or HIBE. We omit the formal definitions, which may be found in the literature [11, 1].

Confidentiality: This is the usual security notion of semantic security for encryption. It means that no non-trivial information about the message can be feasibly gleaned from the ciphertext.

Anonymity: Recipient anonymity is the property that the adversary be unable to distinguish the encryption of a chosen message for a first chosen identity from the encryption of the same message for a second chosen identity. Equivalently, the adversary must be unable to decide whether a ciphertext was encrypted for a chosen identity, or for a random identity.

3 Intuition

Before we present our scheme we first explain why it is difficult to implement anonymous IBE without random oracles, as well as any form of anonymous HIBE even in the random oracle model. We then give some intuition behind our solution.

3.1 The Difficulty

Recall that in the basic Boneh-Franklin IBE system [11], an encryption of a message Msg to some identity Id, takes the following form,

$$\mathsf{CT} = [\ C_1,\ C_2\ ] = [\ g^r,\ \mathbf{e}(\mathcal{H}(\mathsf{Id}), Q)^r\,\mathsf{Msg}\ ]\ \in\ \mathbb{G} \times \mathbb{G}_T\ ,$$

where $\mathcal{H}$ is a random oracle, r is a random exponent, and g and Q are public system parameters. A crucial observation is that the one element of the ciphertext in the bilinear group $\mathbb{G}$, namely, g^r, is just a random element that gives no information about the identity of the recipient. The reason why only one element in $\mathbb{G}$ is needed is because private keys in the Boneh-Franklin scheme are deterministic — there will be no randomness in the private key to cancel out. Since the proof of semantic security is based on the fact that C_2 is indistinguishable from random without the private key for ID, it follows that the scheme is also anonymous since C_2 is the only part of the ciphertext on which the recipient identity has any bearing.

More recently, there have been a number of IBE schemes proven secure without random oracles, such as BTE from [15], BB_1 and BB_2 from [5], and Waters' [32]. However, in all these schemes the proof of security requires that randomness be injected into the private key generation. Since the private keys are randomized, some extra information is needed in the ciphertext in order to cancel out the randomness upon decryption. To illustrate, consider the encryption of a message Msg to an identity Id in the BB_1 Boneh-Boyen system,

$$\mathsf{CT} = [\ C_1,\ C_2,\ C_3\ ] = \Big[\ g^r,\ (g_1^{\mathsf{Id}}\, g_3)^r,\ \mathbf{e}(g_1, \hat{g}_2)^r\,\mathsf{Msg}\ \Big]\ \in\ \mathbb{G}^2 \times \mathbb{G}_T\ ,$$

where r is chosen by the encryptor and g, g_1, g_3, and $\mathbf{e}(g_1, \hat{g}_2)$ are public system parameters. Notice, there are now two elements in $\mathbb{G}$, and between them there is enough redundancy to determine whether a ciphertext was intended for a given identity Id, simply by testing whether the tuple $[g, g_1^{\mathsf{Id}}\, g_3, C_1, C_2]$ is Diffie-Hellman, using the bilinear map,

$$\mathbf{e}(C_1, \hat{g}_1^{\mathsf{Id}}\, \hat{g}_3) \stackrel{?}{=} \mathbf{e}(C_2, \hat{g})\ .$$

We see that the extra ciphertext components which are seemingly necessary in IBE schemes without random oracles, in fact contribute to leaking the identity of the intended recipient of a ciphertext.

A similar argument can be made for why none of the existing HIBE schemes is anonymous, even though some of them use random oracles. Indeed, all known HIBE schemes, including the Gentry-Silverberg system in the random oracle model, rely on randomization in order to properly delegate private keys down the hierarchy in a collusion-resistant manner. Since the randomization is performed not just by the master authority, but by anyone who has the power to delegate a key, the elements needed for it are distributed as part of the public parameters. Because of this, we end up in the same situation as above, where the extra components needed to either perform or cancel the randomization will also provide a test for the addressee's identity.

Since having randomized keys seems to be fundamental to designing (H)IBE systems without random oracles, we aim to design a system where the necessary extra information will be hidden to a computationally bounded adversary. Thus, even though we cannot prevent the ciphertext from containing information about the recipient, we can design our system such that this information cannot be easily tested from the public parameters and ciphertext alone.

3.2 Our Approach

As mentioned in the introduction, we can prevent a single-level identity to be testable by performing some sort of blinding, by splitting the identity into two randomized complementary components. Indeed, building a "flat" anonymous IBE system turns out to be reasonably straightforward using our linear splitting technique to hide the recipient identity behind some randomization.

Complications arise when one tries to support hierarchical key generation. In a nutshell, to prevent collusion attacks in HIBE, "parents" must independently re-randomize the private keys they give to their "children". In all known HIBE schemes, re-randomization is enabled by listing a number of supplemental components in the public system parameters. Why this breaks anonymity is because the same mechanism that allows private keys to be publicly re-randomized, also allows ciphertexts to be publicly tested for recipient identities. Random oracles offer no protection against this.

To circumvent this obstable, we need to make the re-randomization elements non-public, and tie them to each individual private key. In practical terms, this means that private keys must convey extra components (although not too many). The real difficulty is that each set of re-randomization components constitutes a full-fledged HIBE in its own right, which must be simulated together with its peers in the security proof (their number grows linearly with the maximal depth). Because these systems are not independent but interact with each other, we are left with the task of simulating multiple HIBE subsystems that are globally constrained by a set of linear relations. A novelty of our proof technique is a method to endow the simulator with enough degrees of freedom to reduce a system of unknown keys to a single instance of the presumed hard problem.

A notable feature of our construction is that it can be implemented using all known instantiations of the bilinear pairing (whether symmetric or asymmetric, with our without a computable or invertible homomorphism, *etc.*). To cover all grounds, we first describe a "flat" anonymous IBE using the symmetric notation, for ease of exposition, and then move to the full HIBE using the general asymmetric notation without assuming any homomorphism, for maximum generality.

4 A Primer : Anonymous IBE

We start by describing an Anonymous IBE scheme that is semantically secure against selective-ID chosen plaintext attacks. This construction will illustrate our basic technique of "splitting" the bilinear group elements into two pieces to protect against the attacks described in the previous section. In the next section we will describe our full Anonymous HIBE scheme, and in the appendix mention how to achieve adaptive-ID and chosen ciphertext security.

For simplicity, and also to show that we get anonymity even when using symmetric pairings, we describe the IBE system (and the IBE system only) in the special case where $\mathbb{G} = \hat{\mathbb{G}}$:

Setup The setup algorithm chooses a random generator $g \in \mathbb{G}$, random group elements $g_0, g_1 \in \mathbb{G}$, and random exponents $\omega, t_1, t_2, t_3, t_4 \in \mathbb{Z}_p$. It keeps these exponents as the master key, Msk. The corresponding system parameters are published as:

$$\mathsf{Pub} \leftarrow \left[\ \Omega = \mathbf{e}(g,g)^{t_1 t_2 \omega},\ g,\ g_0,\ g_1,\ v_1 = g^{t_1},\ v_2 = g^{t_2},\ v_3 = g^{t_3},\ v_4 = g^{t_4}\ \right] .$$

Extract$(\mathsf{Msk}, \mathsf{Id})$ To issue a private key for identity Id, the key extraction authority chooses two random exponents $r_1, r_2 \in \mathbb{Z}_p$, and computes the private key, $\mathsf{Pvk}_{\mathsf{Id}} = [d_0, d_1, d_2, d_3, d_4]$, as:

$$\mathsf{Pvk}_{\mathsf{Id}} \leftarrow \left[\ g^{r_1 t_1 t_2 + r_2 t_3 t_4},\ g^{-\omega t_2}(g_0 g_1^{\mathsf{Id}})^{-r_1 t_2},\ g^{-\omega t_1}(g_0 g_1^{\mathsf{Id}})^{-r_1 t_1},\ (g_0 g_1^{\mathsf{Id}})^{-r_2 t_4},\ (g_0 g_1^{\mathsf{Id}})^{-r_2 t_3}\ \right] .$$

Encrypt$(\mathsf{Pub}, \mathsf{Id}, M)$ Encrypting a message $\mathsf{Msg} \in \mathbb{G}_T$ for an identity $\mathsf{Id} \in \mathbb{Z}_p^\times$ works as follows. The algorithm chooses random exponents $s, s_1, s_2 \in \mathbb{Z}_p$, and creates the ciphertext as:

$$\mathsf{CT}\ = [\ C', C_0, C_1, C_2, C_3, C_4\ ] \leftarrow \left[\ \Omega^s M,\ (g_0 g_1^{\mathsf{Id}})^s,\ v_1^{s-s_1},\ v_2^{s_1},\ v_3^{s-s_2},\ v_4^{s_2}\ \right] .$$

Decrypt$(\mathsf{Pvk}_{\mathsf{Id}}, C)$ The decryption algorithm attempts to decrypt a ciphertext CT by computing:

$$C'\,\mathbf{e}(C_0,d_0)\,\mathbf{e}(C_1,d_1)\,\mathbf{e}(C_2,d_2)\,\mathbf{e}(C_3,d_3)\,\mathbf{e}(C_4,d_4) = \mathsf{Msg} .$$

Proving Security. We prove security using a hybrid experiment. Let $[C', C_0, C_1, C_2, C_3, C_4]$ denote the challenge ciphertext given to the adversary during a real attack. Additionally, let R be a random element of $\mathbb{G}_T$, and R', R'' be random elements of $\mathbb{G}$. We define the following hybrid games which differ on what challenge ciphertext is given by the simulator to the adversary:

Γ_0 : The challenge ciphertext is $\mathsf{CT}_0 = [C', C_0, C_1, C_2, C_3, C_4]$.

Γ_1 : The challenge ciphertext is $\mathsf{CT}_1 = [R, C_0, C_1, C_2, C_3, C_4]$.

Γ_2 : The challenge ciphertext is $\mathsf{CT}_2 = [R, C_0, R', C_2, C_3, C_4]$.

Γ_3 : The challenge ciphertext is $\mathsf{CT}_3 = [R, C_0, R', C_2, R'', C_4]$.

We remark that the challenge ciphertext in Γ_3 leaks no information about the identity since it is composed of six random group elements, whereas in Γ_0 the challenge is well formed. We show that the transitions from Γ_0 to Γ_1 to Γ_2 to Γ_3 are all computationally indistinguishable.

Lemma 1 (semantic security). *Under the (t, ϵ)-Decision BDH assumption, there is no adversary running in time t that distinguishes between the games Γ_0 and Γ_1 with advantage greater than ϵ.*

Proof. The proof from this lemma essentially follows from the security of the Boneh-Boyen selective-ID scheme. Suppose there is an adversary that can distinguish between game Γ_0 and Γ_1 with advantage ϵ. Then we build a simulator that plays the Decision BDH game with advantage ϵ.

The simulator receives a D-BDH challenge $[g, g^{z_1}, g^{z_2}, g^{z_3}, Z]$ where Z is either $\mathbf{e}(g,g)^{z_1 z_2 z_3}$ or a random element of $\mathbb{G}_T$ with equal probability. The game proceeds as follows:

⋄ *Init:* The adversary announces the identity Id^* it wants to be challenged upon.

⋄ *Setup:* The simulator chooses random exponents $t_1, t_2, t_3, t_4, y \in \mathbb{Z}_p$. It retains the generator g, and sets $g_0 = (g^{z_1})^{-\mathsf{Id}^*} g^y$ and $g_1 = g^{z_1}$. The public parameters are published as:

$$\mathsf{Pub} \leftarrow \left[\ \Omega = \mathbf{e}(g^{z_1}, g^{z_2})^{t_1 t_2},\ g,\ g_0,\ g_1,\ v_1 = g^{t_1},\ v_2 = g^{t_2},\ v_3 = g^{t_3},\ v_4 = g^{t_4}\ \right] .$$

Note that this implies that $\omega = z_1 z_2$.

⋄ *Phase 1:* Suppose the adversary requests a key for identity $\mathsf{Id} \neq \mathsf{Id}^*$. The simulator picks random exponents $r_1, r_2 \in \mathbb{Z}_p$, and issues a private key as: $\mathsf{Pvk}_{\mathsf{Id}} = [d_0, d_1, d_2, d_3, d_4] \leftarrow$

$$\left[\ (g^{z_2})^{\frac{-1}{\mathsf{Id}-\mathsf{Id}^*}} g^{r_1} g^{r_2 t_3 t_4},\ ((g^{z_2})^{\frac{y}{\mathsf{Id}-\mathsf{Id}^*}} (g_0 g_1^{\mathsf{Id}})^{r_1})^{-t_2},\ ((g^{z_2})^{\frac{y}{\mathsf{Id}-\mathsf{Id}^*}} (g_0 g_1^{\mathsf{Id}})^{r_1})^{-t_1},\ (g_0 g_1^{\mathsf{Id}})^{-r_2 t_4},\ (g_0 g_1^{\mathsf{Id}})^{-r_2 t_3}\ \right] .$$

This is a well formed secret key for random exponents $\tilde{r_1} = r_1 - z_2/(\mathsf{Id} - \mathsf{Id}^*)$ and $\tilde{r_2} = r_2$.

⋄ *Challenge:* Upon receiving a message Msg from the adversary, the simulator chooses $s_1, s_2 \in \mathbb{Z}_p$, and outputs the challenge ciphertext as:

$$\mathsf{CT}\ = [\ C', C_0, C_1, C_2, C_3, C_4\ ] \leftarrow \left[\ Z^{-t_1 t_2} M,\ (g^{z_3})^y,\ (g^{z_3})^{t_1} g^{-s_1 t_1},\ g^{s_1 t_2},\ (g^{z_3})^{t_3} g^{-s_2 t_3},\ g^{s_2 t_4}\ \right] .$$

We can let $s = z_3$ and see that if $Z = \mathbf{e}(g,g)^{z_1 z_2 z_3}$ the simulator is playing game Γ_0 with the adversary, otherwise the simulator is playing game Γ_1 with the adversary.

⋄ *Phase 2:* The simulator answers the queries in the same way as Phase 1.

⋄ *Guess:* The simulator outputs a guess γ, which the simulator forwards as its own guess for the D-BDH game.

Since the simulator plays game Γ_0 if and only the given D-BDH instance was well formed, the simulator's advantage in the D-BDH game is exactly ϵ. □

Lemma 2 (anonymity, part 1). *Under the (t,ϵ)-Decision linear assumption, no adversary that runs in time t can distinguish between the games Γ_1 and Γ_2 with advantage greater than ϵ.*

Proof. Suppose the existence of an adversary $\mathcal{A}$ that distinguishes between the two games with advantage ϵ. Then we construct a simulator that wins the Decision Linear game as follows.

The simulator takes in a D-Linear instance $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, Z]$, where Z is either $g^{z_3+z_4}$ or random in $\mathbb{G}$ with equal probability. For convenience, we rewrite this as $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, Y, g^s]$ for s such that $g^s = Z$, and consider the task of deciding whether $Y = g^{z_2(s-z_3)}$ which is equivalent. The simulator plays the game in the following stages.

⋄ *Init:* The adversary $\mathcal{A}$ gives the simulator the challenge identity Id^*.

⋄ *Setup:* The simulator first chooses random exponents $\alpha, y, t_3, t_4, \omega$. It lets g in the simulation be as in the instance, and sets $v_1 = g^{z_2}$ and $v_2 = g^{z_1}$. The public key is published as: $\mathsf{Pub} \leftarrow$

$$\left[\ \Omega = \mathbf{e}(g^{z_1}, g^{z_2})^{\omega},\ g,\ g_0 = (g^{z_2})^{-\mathsf{Id}^* \alpha} g^y,\ g_1 = (g^{z_2})^{\alpha},\ v_1 = (g^{z_2}),\ v_2 = (g^{z_1}),\ v_3 = g^{t_3},\ v_4 = g^{t_4}\ \right] .$$

If we pose $t_1 = z_2$ and $t_2 = z_1$, we note that the public key is distributed as in the real scheme.

$\diamond$ *Phase 1:* To answer a private key extraction query for an identity $\mathsf{Id} \neq \mathsf{Id}^*$, the simulator chooses random exponents $r_1, r_2 \in \mathbb{Z}_p$, and outputs a key given by: $\mathsf{Pvk}_{\mathsf{Id}} = [d_0, d_1, d_2, d_3, d_4] \leftarrow$

$$\left[(g^{z_1})^{r_1} g^{r_2 t_3 t_4},\ (g^{z_1})^{-\omega - \alpha(\mathsf{Id}-\mathsf{Id}^*) r_1},\ (g^{z_2})^{-\omega - \alpha(\mathsf{Id}-\mathsf{Id}^*) r_1},\ (g^{z_1})^{\frac{-r_1 y}{t_3}} (g_0 g_1^{\mathsf{Id}})^{-r_2 t_4},\ (g^{z_1})^{\frac{-r_1 y}{t_4}} (g_0 g_1^{\mathsf{Id}})^{-r_2 t_3} \right] .$$

If, instead of r_1 and r_2, we consider this pair of uniform random exponents,

$$\tilde{r_1} = \frac{r_1 \alpha(\mathsf{Id} - \mathsf{Id}^*)}{\alpha(\mathsf{Id} - \mathsf{Id}^*) z_2 + y} , \qquad \tilde{r_2} = r_2 + \frac{y z_1 r_1}{(t_3 t_4)(\alpha(\mathsf{Id} - \mathsf{Id}^*) z_2 + y)} ,$$

then we see that the private key is well formed, since it can be rewritten as:

$$\mathsf{Pvk}_{\mathsf{Id}} = \left[g^{\tilde{r_1} t_1 t_2 + \tilde{r_2} t_3 t_4},\ g^{-\omega t_2} (g_0 g_1^{\mathsf{Id}})^{-\tilde{r_1} t_2},\ g^{-\omega t_1} (g_0 g_1^{\mathsf{Id}})^{-\tilde{r_1} t_1},\ (g_0 g_1^{\mathsf{Id}})^{-\tilde{r_2} t_4},\ (g_0 g_1^{\mathsf{Id}})^{-\tilde{r_2} t_3} \right] .$$

$\diamond$ *Challenge:* The simulator gets from the adversary a message M which it can discard, and responds with a challenge ciphertext for the identity Id^*. Pose $s_1 = z_3$. To proceed, the simulator picks a random exponent $s_2 \in \mathbb{Z}_p$ and a random element $R \in \mathbb{G}_T$, and outputs the ciphertext as:

$$\mathsf{CT} = [\, C', C_0, C_1, C_2, C_3, C_4 \,] \leftarrow \left[R,\ (g^s)^y,\ Y,\ (g^{z_1 z_3}),\ (g^s)^{t_3} g^{-s_2 t_3},\ g^{s_2 t_4} \right] .$$

If $Y = g^{z_2(s - z_3)}$, *i.e.*, $g^s = Z = g^{z_3 + z_4}$, then $C_1 = v_1^{s - s_1}$ and $C_2 = v_2^{s_1}$; all parts of the challenge but C' are thus well formed, and the simulator behaved as in game Γ_1. If instead Y is independent of z_1, z_2, s, s_1, s_2, which happens when Z is random, then the simulator responded as in game Γ_2.

$\diamond$ *Phase 2:* The simulator answer the query in the same way as Phase 1.

$\diamond$ *Output:* The adversary outputs a bit γ to guess which hybrid game the simulator has been playing. To conclude, the simulator forwards γ as its own answer in the Decision-Linear game.

By the simulation setup the advantage of the simulator will be exactly that of the adversary. $\square$

Lemma 3 (anonymity, part 2). *Under the (t, ϵ)-Decision linear assumption, no adversary that runs in time t can distinguish between the games Γ_2 and Γ_3 with advantage greater than ϵ.*

Proof. This argument follows almost identically to that of Lemma 2, except where the simulation is done over the parameters v_3 and v_4 in place of v_1 and v_2. The other difference is that the g^ω term that appeared in d_1, d_2 without interfering with the simulation, does not even appear in d_3, d_4. $\square$

5 The Scheme : Anonymous HIBE

We now describe our full Anonymous HIBE scheme without random oracles. Anonymity is provided by the splitting technique and hybrid proof introduced in the previous section. In addition, to thwart the multiple avenues for user collusion enabled by the hierarchy, the keys are re-randomized between all siblings and all children. Roughly speaking, this is done by using several parallel HIBE systems, which are recombined at random every time a new private key is issued. In the proof of security, this extra complication is handled by a "multi-secret simulator", that is able to simulate multiple interacting HIBE systems under a set of constraints. This is an information theoretic proof that sits on top of the hybrid argument, which is computational.

For the most part, we focus on security against selective-identity, chosen plaintext attacks. In Appendix A we mention how to secure the scheme against adaptive-ID and CCA2 adversaries.

Setup$(1^\Sigma, D)$ To generate the public system parameters and the corresponding master secret key, given a security parameter $\Sigma \in \mathbb{N}$ in unary, and the hierarchy's maximum depth $D \in \mathbb{N}$, the setup algorithm first generates a bilinear instance $\mathbf{G} = [p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, g, \hat{g}, \mathbf{e}] \xleftarrow{\$} \mathrm{Gen}(1^\Sigma)$. Then:

1. Select $7 + 5\,D + D^2$ random integers modulo p (some of them forcibly non-zero):

$$\omega,\ [\ \alpha_n,\ \beta_n,\ [\ \theta_{n,\ell}\ ]_{\ell=0,\dots,D}\ ]_{n=0,\dots,1+D} \quad \in_\$ \mathbb{Z}_p^\times \times ((\mathbb{Z}_p^\times)^2 \times (\mathbb{Z}_p)^{1+D})^{2+D}\ ;$$

2. Publish $\mathbf{G}$ and the system parameters $\mathsf{Pub} \in \mathbb{G}_T \times \mathbb{G}^{2\,(1+D)\,(2+D)}$ given by:

$$\Omega,\ [\ [\ a_{n,\ell},\ b_{n,\ell}\ ]_{\ell=0,\dots,D}\ ]_{n=0,\dots,1+D} \ \leftarrow\ \mathbf{e}(g,\hat{g})^\omega,\ \Big[\ [\ g^{\alpha_n\,\theta_{n,\ell}},\ g^{\beta_n\,\theta_{n,\ell}}\ ]_{\ell=0,\dots,D}\ \Big]_{n=0,\dots,1+D}$$

3. Retain the master secret key $\mathsf{Msk} \in \hat{\mathbb{G}}^{1+(3+D)\,(2+D)}$ comprising the elements:

$$\hat{w},\ [\ \hat{a}_n,\ \hat{b}_n,\ [\ \hat{y}_{n,\ell}\ ]_{\ell=0,\dots,D}\ ]_{n=0,\dots,1+D} \ \leftarrow\ \hat{g}^\omega,\ \begin{bmatrix} \hat{g}^{\alpha_n},\ \hat{g}^{\beta_n}, \\ [\ \hat{g}^{\alpha_n\,\beta_n\,\theta_{n,\ell}}\ ]_{\ell=0,\dots,D} \end{bmatrix}_{n=0,\dots,1+D} .$$

Extract$(\mathsf{Pub}, \mathsf{Msk}, \mathsf{Id})$ To extract a private key for an identity $\mathsf{Id} = [I_0, I_1, \dots, I_L] \in (\mathbb{Z}_p^\times)^{1+L}$ where $L \in \{1, \dots, D\}$ and by convention $I_0 = 1$, using the master key Msk:

1. Pick $6 + 5\,D + D^2$ random integers: $[\rho_n, [\rho_{n,m}]_{m=0,\dots,1+D}]_{n=0,\dots,1+D} \in_\$ (\mathbb{Z}_p)^{(3+D)\,(2+D)}$.
2. Compute the key's decryption portion: $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}} = k_0, [k_{n,(a)}, k_{n,(b)}]_{n=0,\dots,1+D} \leftarrow$

$$\hat{w} \prod_{n=0}^{1+D} \prod_{\ell=0}^{L} (\hat{y}_{n,\ell}^{I_\ell})^{\rho_n},\ \Big[\ \hat{a}_n^{-\rho_n},\ \hat{b}_n^{-\rho_n}\ \Big]_{n=0,\dots,1+D} \quad \in\ \hat{\mathbb{G}}^{5+2\,D}\ .$$

3. The re-randomization part: $\mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}} = [f_{m,0}, [f_{m,n,(a)}, f_{m,n,(b)}]_{n=0,\dots,1+D}]_{m=0,\dots,1+D} \leftarrow$

$$\left[\prod_{n=0}^{1+D} \prod_{\ell=0}^{L} (\hat{y}_{n,\ell}^{I_\ell})^{\rho_{n,m}},\ \Big[\ \hat{a}_n^{-\rho_{n,m}},\ \hat{b}_n^{-\rho_{n,m}}\ \Big]_{n=0,\dots,1+D} \right]_{m=0,\dots,1+D} \quad \in\ \hat{\mathbb{G}}^{(5+2\,D)\,(2+D)}\ .$$

4. And then the delegation components: $\mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}} = [h_\ell, [h_{m,\ell}]_{m=0,\dots,1+D}]_{\ell=1+L,\dots,D} \leftarrow$

$$\left[\prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_n},\ \left[\prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_{n,m}} \right]_{m=0,\dots,1+D} \right]_{\ell=1+L,\dots,D} \quad \in\ \hat{\mathbb{G}}^{(3+D)\,(D-L)}\ .$$

The full private key is issued as the concatenation: $\mathsf{Pvk}_{\mathsf{Id}} = \mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}} \,\|\, \mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}} \,\|\, \mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}}$.

A more intuitive way to visualize the private key is as a rectangular array in $\hat{\mathbb{G}}^{(3+D)\times(5+3\,D-L)}$ with $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}}$ in the upper left corner, $\mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}}$ in the lower left, and $\mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}}$ on the right side:

$$\begin{matrix} \begin{bmatrix} k_0 & k_{1,(a)} & k_{1,(b)} & \dots & k_{1+D,(a)} & k_{1+D,(b)} \end{bmatrix} \\ \begin{bmatrix} f_{0,0} & f_{0,0,(a)} & f_{0,0,(a)} & \dots & f_{0,1+D,(a)} & f_{0,1+D,(a)} \\ f_{1,0} & f_{1,0,(a)} & f_{1,0,(a)} & \dots & f_{1,1+D,(a)} & f_{1,1+D,(a)} \\ \vdots & & & \ddots & & \\ f_{1+D,0} & f_{1+D,0,(a)} & f_{1+D,0,(a)} & \dots & f_{1+D,1+D,(a)} & f_{1+D,1+D,(a)} \end{bmatrix} \end{matrix} \begin{bmatrix} h_{1+L} & h_{2+L} & \dots & h_D \\ h_{0,1+L} & h_{0,2+L} & \dots & h_{0,D} \\ h_{1,1+L} & h_{1,2+L} & \dots & h_{1,D} \\ & & \ddots & \\ h_{1+D,1+L} & h_{1+D,2+L} & \dots & h_{1+D,D} \end{bmatrix}$$

Each row on the left can be viewed as a private key in an independent HIBE system (with generalized linear splitting as in Section 4). The main difference is that only $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}}$ contains the secret $\hat{w}$. The rows of $\mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}}$ are independent HIBE keys for the same Id that do not permit decryption. The elements on the right side provide the delegation functionality: each column in $\mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}}$ extends the hierarchy down one level. Delegation works as follows:

Derive$(\mathsf{Pub}, \mathsf{Pvk}_{\mathsf{Id}|L-1}, \mathsf{Id}_L)$ This algorithm derives a private key for $\mathsf{Id} = [I_0, I_1, \ldots, I_L] \in (\mathbb{Z}_p^\times)^{1+L}$ where $L \in \{2, \ldots, D\}$ and $I_0 = 1$, given a private key of the parent. Let that be $\mathsf{Pvk}_{\mathsf{Id}|L-1} = [k_0, [k_{n,(a)}, k_{n,(b)}], [f_{m,0}, [f_{m,n,(a)}, f_{m,n,(b)}]], [h_\ell, [h_{m,\ell}]]_{\ell=L,\ldots,D}]$ for $n, m \in \{0, \ldots, 1+D\}$.

1. Pick $6+5\,D+D^2$ random integers: $[\pi_m, [\pi_{m,m'}]_{m'=0,\ldots,1+D}]_{m=0,\ldots,1+D} \in_\$ (\mathbb{Z}_p)^{(3+D)\,(2+D)}$.
2. Compute for the decryption portion: $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}} = k'_0, [k'_{n,(a)}, k'_{n,(b)}]_{n=0,\ldots,1+D} \leftarrow$

$$(k_0 \prod_{m=0}^{1+D} (f_{m,0})^{\pi_m})\,(h_\ell \prod_{m=0}^{1+D} (h_{m,\ell})^{\pi_m})^{I_L},\ \left[\ k_{n,(a)} \prod_{m=0}^{1+D} (f_{m,n,(a)})^{\pi_m},\quad k_{n,(b)} \prod_{m=0}^{1+D} (f_{m,n,(b)})^{\pi_m}\ \right]_{n=0,\ldots,1+D}.$$

3. For re-randomization: $\mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}} = [f'_{m',0}, [f'_{m',n,(a)}, f'_{m',n,(b)}]_{n=0,\ldots,1+D}]_{m'=0,\ldots,1+D} \leftarrow$

$$\left[\ (\prod_{m=0}^{1+D} (f_{m,0})^{\pi_{m,m'}})\,(\prod_{m=0}^{1+D} (h_{m,\ell})^{\pi_{m,m'}})^{I_L},\ \left[\ \prod_{m=0}^{1+D} (f_{m,n,(a)})^{\pi_{m,m'}},\ \prod_{m=0}^{1+D} (f_{m,n,(b)})^{\pi_{m,m'}}\ \right]_{n=0,\ldots,1+D}\ \right]_{m'=0,\ldots,1+D}.$$

4. And then for delegation: $\mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}} = [h'_\ell, [h'_{m',\ell}]_{m'=0,\ldots,1+D}]_{\ell=1+L,\ldots,D} \leftarrow$

$$\left[\ h_\ell \prod_{m=0}^{1+D} (h_{m,\ell})^{\pi_m},\quad \left[\ \prod_{m=0}^{1+D} (h_{m,\ell})^{\pi_{m,m'}}\ \right]_{m'=0,\ldots,1+D}\ \right]_{\ell=1+L,\ldots,D}.$$

The subordinate private key is the concatenation: $\mathsf{Pvk}_{\mathsf{Id}} = \mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}} \parallel \mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}} \parallel \mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}}$.

Derive and *Extract* create private keys with the same structure and distribution. The derivation process in *Derive* merges two distinct operations: delegation and re-randomization.
– Re-randomization occurs first, conceptually speaking. Very simply, we take a random linear combination of all the rows of the big array on page 11. The first row is treated a bit differently: it does not intervene into any other row's re-randomization, and its own coefficient is set to 1.
– Delegation targets the leftmost elements of $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}}$ and $\mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}}$, where identities appear. Imagine $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}}$, $\mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}}$, and $\mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}}$ after re-randomization. Delegation to sub-identity I_L "consumes" the first column of $\mathsf{Pvk}_{\mathsf{Id}}^{\text{deleg}}$: each element is raised to the power of I_L, and the result is aggregated into its target, the leftmost element of $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}}$ or $\mathsf{Pvk}_{\mathsf{Id}}^{\text{rerand}}$ on the same row:

$$\begin{matrix}[\ k_0 & \cdots & k_{n,(a)} & k_{n,(b)} & \cdots\ ] \\ \begin{bmatrix} f_{0,0} & \cdots & f_{0,n,(a)} & f_{0,n,(b)} & \cdots \\ \vdots & \ddots & & & \\ f_{m,0} & \cdots & f_{m,n,(a)} & f_{m,n,(b)} & \cdots \\ \vdots & & \ddots & & \end{bmatrix}\end{matrix} \begin{bmatrix} h_L & h_{1+L} & \cdots & h_D \\ h_{0,L} & h_{0,1+L} & \cdots & h_{0,D} \\ \vdots & \ddots & & \\ h_{m,L} & h_{m,1+L} & \cdots & h_{m,D} \\ \vdots & & & \ddots \end{bmatrix} \to \begin{matrix}[\ k'_0 & \cdots\ ] \\ \begin{bmatrix} f'_{0,0} & \cdots \\ \vdots & \ddots \\ f'_{m,0} & \cdots \\ \vdots & \end{bmatrix}\end{matrix} \begin{bmatrix} \bullet & h'_{1+L} & \cdots & h'_D \\ \bullet & h'_{0,1+L} & \cdots & h'_{0,D} \\ & \ddots & & \\ \bullet & h'_{m,1+L} & \cdots & h'_{m,D} \\ & & & \ddots \end{bmatrix}$$

We now turn to the encryption and decryption methods.

Encrypt(Pub, Id, Msg) To encrypt a message encoded as a group element $\mathsf{Msg} \in \mathbb{G}_T$ for a given identity $\mathsf{Id} = [I_0(=1), I_1, \ldots, I_L]$ at level L, the encryption algorithm proceeds as follows:

1. Select $3 + D$ random integers: $r, [r_n]_{n=0,\ldots,1+D} \in_{\$} (\mathbb{Z}_p)^{3+D}$.
2. Output the ciphertext: $\mathsf{CT} = E, c_0, [c_{n,(a)}, c_{n,(b)}]_{n=0,\ldots,1+D} \leftarrow$

$$\mathsf{Msg} \cdot \Omega^{-r}, \quad g^r, \quad \left[\left(\prod_{\ell=0}^{L} b_{n,\ell}^{I_\ell} \right)^{r_n}, \left(\prod_{\ell=0}^{L} a_{n,\ell}^{I_\ell} \right)^{r-r_n} \right]_{n=0,\ldots,1+D} \quad \in \mathbb{G}_T \times \mathbb{G}^{5+2\,D} .$$

Encryption is very cheap with a bit of caching since the exponentiation bases never change.

Decrypt($\mathsf{Pub}, \mathsf{Pvk}_{\mathsf{Id}}, \mathsf{CT}$) To decrypt a ciphertext CT, using (the decryption portion of) a private key $\mathsf{Pvk}_{\mathsf{Id}}^{\text{decrypt}} = [k_0, [k_{n,(a)}, k_{n,(b)}]_{n=0,\ldots,1+D}]$, the decryption algorithm outputs:

$$\hat{\mathsf{Msg}} \leftarrow E \cdot \mathbf{e}(c_0, k_0) \prod_{n=0}^{1+D} \mathbf{e}(c_{n,(a)}, k_{n,(a)})\, \mathbf{e}(c_{n,(b)}, k_{n,(b)}) \in \mathbb{G}_T .$$

All the pairings in the product can be computed at once using a "multi-pairing" approach [22], which is similar to multi-exponentiation. One can also exploit the fact that all the $k_{\ldots}$ are fixed for a given recipient to perform advantageous pre-computations [3].

6 Properties

We formally state the main properties of the A-HIBE scheme given in the previous section, and discuss how to achieve security against active attacks.

6.1 Consistency

The following theorems show that extracted and delegated private keys are identically distributed, and that extraction, encryption, and decryption, are consistent. We remark that Theorem 4 is not essential for the security model, but it is nice to have and it is also useful to prove Theorem 5. Proofs are given in Appendix B.

Theorem 4. *Private keys calculated by* Derive *and* Extract *have the same distribution.*

Theorem 5. *The Anonymous HIBE scheme is internally consistent.*

Proofs. Detailed proofs of these theorems are given in Appendix B. □

6.2 Security

We now state the basic security theorems for the A-HIBE scheme. The selective-ID security reductions are almost tight and hold in the standard model. We only consider recipient anonymity, since sender anonymity is trivially attainable in an unauthenticated encryption scheme. Informal arguments and full proofs may be found in Appendix C.

Theorem 6 (Confidentiality). *Suppose that* $\mathbf{G}$ *upholds the* (τ, ϵ)*-Decision BDH assumption. Then, against a selective-ID adversary that makes at most* q *private key extraction queries, the HIBE scheme of Section 5 is* $(q, \tilde{\tau}, \tilde{\epsilon})$*-IND-sID-CPA secure in* $\mathbf{G}$ *with* $\tilde{\tau} \approx \tau$ *and* $\tilde{\epsilon} = \epsilon - (3 + D)\, q/p$.

Theorem 7 (Anonymity). *Suppose that* $\mathbf{G}$ *upholds the* (τ, ϵ)*-Decision Linear assumption. Then, against a selective-ID adversary that makes* q *private key extraction queries, the HIBE scheme of Section 5 is* $(q, \tilde{\tau}, \tilde{\epsilon})$*-ANON-sID-CPA secure in* $\mathbf{G}$ *with* $\tilde{\tau} \approx \tau$ *and* $\tilde{\epsilon} = \epsilon - (2 + D)\,(7 + 3\,D)\, q/p$.

Proofs. Detailed proofs of these theorems are given in Appendices C.1 and C.2. □

6.3 Active Adversaries

We briefly mention how to strengthen the scheme against active attacks. See also Appendix A.

Adaptive-ID Security. Security against "full" adaptive-identity attacks in the standard model can be achieved in a number of ways. The most elegant uses Waters [32] hashing technique, originally proposed for the BB_1 scheme, but which is also applicable here. Alternatively, we may use one of the two transformations described in [5, §7]; these are simple and generic (one of them uses random oracles), but incur a substantial security degradation, which must be compensated by increasing the size of the bilinear groups for a fixed security level.

None of these methods is very satisfactory in the HIBE case, because security always degrades exponentially with the depth of the hierarchy. In other words, to achieve any desired final bit strength, we need to increase the bilinear group bit size, $\log_2(p)$, linearly with the depth of the hieararchy. Practically, this makes the system more computationally demanding. Theoretically, this means that the adaptive-ID security reduction is not polynomial in the depth of the hierarchy. Indeed, this is symptomatic of the fact that none of HIBE schemes known to date is "asymptotically secure" (*i.e.*, with a polynomial security reduction in all parameters), in the adaptive-ID security model, even among those that rely on random oracles.

CCA2 Security. Several efficient transformations from CPA-secure HIBE to CCA2-secure HIBE can be used to obtain CCA2 security. The original method of Canetti, Halevi, and Katz [16] and the more efficient version due to Boneh and Katz [12] are generic. The method of Boyen, Mei, and Waters [14] is not, but is compatible with the present scheme and is slightly more economical. None of these transformations will have an adverse impact on anonymity.

7 Conclusion

We presented a provably anonymous IBE and HIBE scheme without random oracles, which resolves an open question from CRYPTO 2005 regarding the existence of anonymous HIBE systems.

Our constructions make use of a novel "linear-splitting" technique which prevents an attacker from testing the intended recipient of ciphertexts, yet allows for the use of randomized private IBE keys. In the hierarchical case, we add to this a new "multi-simulation" proof device that permits multiple HIBE subsystems to concurrently re-randomize each other. Security is based solely on the Linear assumption in bilinear groups.

Our basic scheme is very efficient, a factor two slower than (non-anonymous) Boneh-Boyen BB_1 and BB_2 encryption, and quite faster than Boneh-Franklin. The full hierarchical scheme remains practical with its quadratic private key size, and its linear ciphertext size, encryption time, and decryption time, as functions of the depth of the hierarchy.

Acknowledgements

The authors would like to thank Mihir Bellare, Dan Boneh, and Hovav Shacham for helpful discussions, as well as the anonymous referees for useful comments.

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 *Advances in Cryptology—CRYPTO 2005*, Lecture Notes in Computer Science, pages 205–22. Springer-Verlag, 2005.

[2] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. Cryptology ePrint Archive, Report 2005/133, 2005. `http://eprint.iacr.org/`.

[3] Paulo S.L.M. Barreto, Hae Y. Kim, Ben Lynn, and Michael Scott. Efficient algorithms for pairing-based cryptosystems. Cryptology ePrint Archive, Report 2002/008, 2002. `http://eprint.iacr.org/`.

[4] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *Proceedings of ASIACRYPT 2001*, Lecture Notes in Computer Science, pages 566–82. Springer-Verlag, 2001.

[5] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–38. Springer-Verlag, 2004.

[6] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–59. Springer-Verlag, 2004.

[7] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–56. Springer-Verlag, 2005.

[8] Dan Boneh, Xavier Boyen, and Shai Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In *Proceedings of RSA-CT 2006*, volume 3860 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006.

[9] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer-Verlag, 2004.

[10] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–22. Springer-Verlag, 2004.

[11] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003. Extended abstract in *Advances in Cryptology—CRYPTO 2001*.

[12] Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In *Proceedings of CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[13] Xavier Boyen. Multipurpose identity-based signcryption: A Swiss Army knife for identity-based cryptography. In *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–99. Springer-Verlag, 2003.

[14] Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM Conference on Computer and Communications Security—CCS 2005*. ACM Press, 2005.

[15] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology—EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.

[16] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–22. Springer-Verlag, 2004.

[17] Sanjit Chatterjee and Palash Sarkar. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In *Proceedings of ICISC 2005*, 2005.

[18] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, 2001.

[19] Darren Davis, Fabian Monrose, and Michael K. Reiter. Time-scoped searching of encrypted audit logs. In *Proceedings of ICICS 2004*, pages 532–45, 2004.

[20] Craig Gentry. Practical identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2006*, Lecture Notes in Computer Science. Springer-Verlag, 2006.

[21] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In *Proceedings of ASIACRYPT 2002*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

[22] Robert Granger and Nigel P. Smart. On computing products of pairings. Cryptology ePrint Archive, Report 2006/172, 2006. `http://eprint.iacr.org/`.

[23] Jeremy Horwitz and Ben Lynn. Towards hierarchical identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2002*, Lecture Notes in Computer Science, pages 466–81. Springer-Verlag, 2002.

[24] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology*, 17(4):263–76, 2004. Extended abstract in *Proceedings of ANTS IV, 2000*.

[25] Alfred Menezes, Tatsuaki Okamoto, and Scott Vanstone. Reducing elliptic curve logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–46, 1993.

[26] Victor Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4), 2004.

[27] David Naccache. Secure and practical identity-based encryption. Cryptology ePrint Archive, Report 2005/369, 2005. `http://eprint.iacr.org/`.

[28] Moni Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer-Verlag, 2003.

[29] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology—CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.

[30] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology—EUROCRYPT 1997*, volume 1233 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[31] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Proceedings of the IEEE Symposium on Security and Privacy—SP 2000*. IEEE Computer Society, 2000.

[32] Brent Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[33] Brent Waters, Dirk Balfanz, Glenn Durfee, and Diana Smetters. Building an encrypted and searchable audit log. In *Proceedings of NDSS 2004*, 2004.

[34] Danfeng Yao, Nelly Fazio, Yevgeniy Dodis, and Anna Lysyanskaya. ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. In *ACM Conference on Computer and Communications Security—CCS 2004*, pages 354–63, 2004.

A Extensions

In this section, we describe a number of interesting extensions that can boost the security and usefulness of our Anonymous HIBE scheme.

Adaptive-ID Security. It is not difficult to modify the algorithms given in Section 5 to achieve provable security against adaptive-identity attacks in the standard model. The generalization is similar to that proposed by Waters and others [32, 27, 17] for the Boneh-Boyen BB_1 scheme [5].

In the selective-ID scheme, each identity component I_ℓ was a single integer in $\mathbb{Z}_p^\times$ (that could result from hashing an identity string using a collision resistant hash function). In the adaptive-ID scheme, we express this identity component as a vector of sub-components which are small integers. In other words, each $I_\ell \in \mathbb{Z}_p^\times$ becomes a vector $\vec{I_\ell} = [I_{\ell,1}, \ldots, I_{\ell,d}] \in \{1, \ldots, R\}^d$ for some small fixed R and d. Essentially, what we have done is to represent the integers I_ℓ as vectors of d digits in radix R. The reasons why this is useful have to do with providing a sparse set of adversarially unpredictable collisions, for the adaptive-ID proof of security, and are discussed in [32, 27, 17].

CCA2 Security. In an HIBE system, whether selective-ID or adaptive-ID, it is very easy to leverage basic CPA security into CCA2 security in a generic manner for the two security goals we care about, IND and ANON. The approach is due to Canetti, Halevi, and Katz [16], and involves adding one level to the HIBE identity hierarchy; the extra identity component at the bottom is then used to protect the rest of the ciphertext against tampering. This can be done either via a signature scheme as originally suggested in [16], a combination of message authentication code and commitment as proposed in [12], or a mere collision-resistant hash function as in [14]. Among these, the CHK method is the most general and versatile, while the BK and BMW approaches are a bit more efficient, and especially the latter for key encapsulation (it is however not generic, but is compatible with our construction).

In all case we end up adding one level to the hierarchy. Fortunately, the added level need not be anonymous, since the "identity" it corresponds to is a function of the ciphertext itself and is independent of sender and recipient; it is also already public. Thus, the extra level can be implemented using a cheaper method, *e.g.*, using one layer of the BB_1 HIBE scheme which will mesh nicely into our A-HIBE construction.

Threshold. It is a known result [8] that non-interactive CCA2 threshold systems are easy to construct from certain identity-based encryption schemes. In a similar vein, it is easy to extend our basic anonymous HIBE to support non-interactive threshold key generation and/or decryption. We refer to [8] for the specifics of the transformation.

Compression. Lastly, we mention a simple optimization of our scheme that gives slightly shorter private keys and ciphertexts. Recall that $\mathsf{Id} = [I_0, \dots, I_L]$ where $I_0 = 1$, so clearly there is no anonymity requirement on I_0. The ramifications of this observation are that it is possible to let the indices n and m range not from 0 but from 1 to $1+D$ in the private keys and the ciphertexts. The identity component $I_0 = 1$ is still present; however, we no longer make any effort to hide it. As a result, the Anonynous IBE ciphertext overhead is reduced from 7 down to 5 elements of $\mathbb{G}$; for Anonymous HIBE of depth D, the overhead is brought down to $3 + 2\,D$ elements of $\mathbb{G}$.

B Consistency Proofs

We now prove the consistency properties of the Anonymous HIBE scheme stated in Section 6.1.

To prove Theorem 5, we need to show that, with respect to the public parameter and the reference key extraction definitions, the mechanisms for key extraction, delegation, encryption, and decryption, are all consistent. It is useful to start with the proof of Theorem 4, *i.e.*, establish that the keys obtained by delegation are "the same" as those created directly from the master secret.

Proof of Theorem 4. We need to show that, for any given Id, private keys produced by *Derive* are distributed identically as those created by *Extract*.

We focus on the decryption, re-randomization, and delegation portions of the private key, one set at a time. The notation is the same as in the scheme description. To show that the decryption portion is correctly distributed, $\forall n \in \{0, \dots, 1+D\}$, it suffices to pose,

$$\rho'_n = \rho_n + \sum_{m=0}^{1+D} \rho_{n,m}\, \pi_m \ .$$

which allows us to rewrite the relevant components as in the reference algorithm,

$$k'_0 = \hat{w} \prod_{n=0}^{1+D} (\prod_{\ell=0}^{L} (\hat{y}_{n,\ell})^{I_\ell})^{\rho'_n} \ , \qquad [\ k'_{n,(a)},\ k'_{n,(b)}\ ] = \left[\ \hat{a}_n^{-\rho'_n},\ \hat{b}_n^{-\rho'_n}\ \right] \ .$$

Similarly, it can be seen that the remainder of the subordinate private key is correctly distributed, as, $\forall m' \in \{0, \dots, 1+D\}, \forall n \in \{0, \dots, 1+D\}$, the substitutions,

$$\rho'_{n,m'} = \sum_{m=0}^{1+D} \rho_{n,m}\, \pi_{m,m'} \ .$$

let us rewrite the re-randomization components in canonical form,

$$f'_{m',0} = \prod_{n=0}^{1+D} (\prod_{\ell=0}^{L} (\hat{y}_{n,\ell})^{I_\ell})^{\rho'_{n,m'}} \ , \qquad [\ f'_{m',n,(a)},\ f'_{m',n,(b)}\ ] = \left[\ \hat{a}_n^{-\rho'_{n,m'}},\ \hat{b}_n^{-\rho'_{n,m'}}\ \right] \ ,$$

as well as the delegation components, $\forall \ell \in \{1+L, \dots, D\}$,

$$h'_\ell = \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho'_n} \ , \qquad h'_{m',\ell} = \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho'_{n,m'}} \ .$$

It follows that private keys produced by *Extract* and *Derive* have the same distribution and can be used indifferently. $\square$

Proof of Theorem 5. To establish the theorem, it suffices to prove that, with respect to the public parameter and the reference key extraction definitions, the mechanisms for key extraction, delegation, encryption, and decryption, are all correct.

Since we have already shown in Theorem 4 that the private keys generated by *Extract* and *Derive* have the same distribution (for a given identity), we only need to consider one type of key. The key specification from *Extract* is the obvious choice.

We show that the *Decrypt* algorithm will successfully decrypt any ciphertext created by the *Encrypt* algorithm for a matching identity. Indeed,

$$\mathsf{Msg} \cdot \Omega^{-r} \cdot \frac{\mathbf{e}(g^r, \hat{w} \prod_{n=0}^{1+D} \prod_{\ell=0}^{L} (\hat{y}_{n,\ell})^{I_\ell \, \rho_n})}{\prod_{n=0}^{1+D} \mathbf{e}((\prod_{\ell=0}^{L} b_{n,\ell}^{I_\ell})^{r_n}, \hat{a}_n^{\rho_n})\, \mathbf{e}((\prod_{\ell=0}^{L} a_{n,\ell}^{I_\ell})^{r-r_n}, \hat{b}_n^{\rho_n})} = \mathsf{Msg} \cdot \Omega^{-r} \cdot \mathbf{e}(g^r, \hat{w}) \ = \mathsf{Msg} \ .$$

In summary, our Anonymous HIBE scheme is consistent, and furthermore *Extract* and *Derive* induce the same distribution over the private keys, as required. □

C Security Proofs

We now turn to the formal proofs of the security theorems stated in Section 6.2.

C.1 Confidentiality

We prove confidentiality (*i.e.*, semantic security) using a reduction from D-BDH. The proof is not unlike that of other HIBE systems (it vaguely resembles Boneh and Boyen's BB_1 scheme), in that we build a simulator that, lacking the component $\hat{w}$ of the master key Msk, is nonetheless able to simulate all private keys except for the challenge identity selected by the adversary (and that identity's ancestors).

There is a novel difficulty, however. Recall from the description of the scheme that a private key consists not only of decryption components $k.$ but also of re-randomization components $f.$ that are essentially the same as the $k.$ with different randomization exponents. So far so good. The problem is that the private key also contains a number of delegation components $h.$, each of which is required to be "compatible" with the $k.$ or $f.$ on the same row (*i.e.*, use the same randomization). As a result, the simulator must simulate not one but many unknown randomization exponents at once, in order to ensure the simultaneous compatibility of all the $h.$.

We solve this problem by introducing extra degrees of freedom in the simulation, in order to "decouple" the various unknowns, and show that there exists an assignment to the extra coefficients that will satisfy the original constraints. This is one of the reason why we need a large enough number of re-randomization rows $f.$, in order to give us enough degrees of freedom to feed the $h.$.

The formal proof follows.

Proof of Theorem 6. We prove the theorem using the usual indistinguishability game.

To show semantic security from the Decision BDH assumption, suppose a D-BDH problem instance is given as a tuple $[g, g^{z_1}, g^{z_3}, \hat{g}, \hat{g}^{z_1}, \hat{g}^{z_2}, Z] \in \mathbb{G}^3 \times \hat{\mathbb{G}}^3 \times \mathbb{G}_T$ for random $[z_1, z_2, z_3] \in (\mathbb{Z}_p)^3$, such that the test element $Z \in \mathbb{G}_T$ is equal either to $\mathbf{e}(g, \hat{g})^{z_1 \, z_2 \, z_3}$ or to $\mathbf{e}(g, \hat{g})^{z_4}$ for random $z_4 \in \mathbb{Z}_p$. For clarity, we rewrite the D-BDH instance supplied to our reduction algorithm, $\mathcal{B}$, as,

$$[g, g_1, g_3, \hat{g}, \hat{g}_1, \hat{g}_2, Z] \ \in \ \mathbb{G}^3 \times \hat{\mathbb{G}}^3 \times \mathbb{G}_T \ .$$

The reduction proceeds as follows.

◇ *Open :*

The adversary $\mathcal{A}$ opens the game by announcing the identity $\mathsf{Id}^* = [I_0^*, I_1^*, \ldots, I_{L^*}^*]$ it wishes to attack, and where $\mathcal{A}$ is allowed to choose the number of hierarchical components, $L^* \in \{1, \ldots, D\}$. The zero-th component is fixed to $I_0^* = 1$.

◇ *Setup :*

To create public parameters, the simulator $\mathcal{B}$ first draws a tuple of random non-zero integers $[\alpha_n, \beta_n]_{n=0,\ldots,1+D} \in_\$ (\mathbb{Z}_p^\times)^{2\,(2+D)}$, as well as, for each $n = 0, \ldots, 1+D$, a vector of pairs of random integers $[\theta_{n,\ell}, \bar{\theta}_{n,\ell}]_{\ell=0,\ldots,D} \in_\$ (\mathbb{Z}_p)^{2\,(1+D)}$, each subject to the constraint that $\sum_{\ell=0}^{L^*} \bar{\theta}_{n,\ell}\, I_\ell^* = 0 \pmod p$. Next, the simulator assigns,

$$\left[\begin{array}{c} \Omega, \\ [\,[\,a_{n,\ell},\ b_{n,\ell}\,]_{\ell=0,\ldots,D}\,]_{n=0,\ldots,1+D} \end{array}\right] \leftarrow \left[\begin{array}{c} \mathbf{e}(g_1, \hat{g}_2) \ \ (= \mathbf{e}(g, \hat{g})^{z_1 z_2}), \\ \left[\,\left[\,(g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\alpha_n},\ (g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\beta_n}\,\right]_{\ell=0,\ldots,D}\,\right]_{n=0,\ldots,1+D} \end{array}\right].$$

The adversary is provided with the public system parameters, Pub, which comprise $\mathbf{G}$ and the elements Ω and $[[a_{n,\ell}, b_{n,\ell}]_{\ell=0,\ldots,D}]_{n=0,\ldots,1+D}$; their distribution is the same as in the real scheme.

To complete the setup, the simulator computes what it can of the private key. Notice that the public parameter simulation pegs the exponent ω from the real scheme to the product of the exponents z_1 and z_2, which are implicitly defined by the Decision Linear instance but unknown to the simulator. $\mathcal{B}$ thus partially computes the master key, Msk, as, (omitting the crossed-out element)

$$\left[\begin{array}{c} \cancel{\hat{w}}, \\ [\,\hat{a}_n,\ \hat{b}_n,\ [\,\hat{y}_{n,\ell}\,]_{\ell=0,\ldots,D}\,]_{n=0,\ldots,1+D} \end{array}\right] \leftarrow \left[\begin{array}{c} \cancel{\hat{g}^{\omega}}, \\ \left[\,\hat{g}^{\alpha_n},\ \hat{g}^{\beta_n},\ [\,(\hat{g}^{\theta_{n,\ell}}\, \hat{g}_1^{\bar{\theta}_{n,\ell}})^{\alpha_n \beta_n}\,]_{\ell=0,\ldots,D}\,\right]_{n=0,\ldots,1+D} \end{array}\right].$$

◇ *Query :*

In the first probing phase, the adversary makes a number of extraction queries on adaptively chosen identities distinct from Id^* and all its prefixes. Suppose $\mathcal{A}$ makes such a query on $\mathsf{Id} = [I_0, \ldots, I_L]$ such that $1 \le L \le D$. To prepare a response, $\mathcal{B}$ starts by determining the identity component of lowest index, L', such that $I_{L'} \neq I_{L'}^*$, letting $L' = L^* + 1$ in the event that Id^* is a prefix of Id. According to the rules of the game, it is always the case that $1 \le L' \le D$. The construction of the private key is a two-step process. In the first step, $\mathcal{B}$ creates a "prototype" private key for $\mathsf{Id}' = [I_0, \ldots, I_L']$; this identity is either equal to or a prefix of Id, but of course not of Id^*. Define $\Theta_n \leftarrow \sum_{\ell=0}^{L'} \theta_{n,\ell}\, I_\ell$ and $\bar{\Theta}_n \leftarrow \sum_{\ell=0}^{L'} \bar{\theta}_{n,\ell}\, I_\ell$ for all $n = 0, \ldots, 1+D$, and note that $(\forall n),\ \bar{\Theta}_n \neq 0 \pmod p$ except with some probability $\le {}^{(2+D)}\!/_p$ over the choice of $[\bar{\theta}_{n,\ell}]$, which is invisible to the adversary. To proceed, $\mathcal{B}$ picks a tuple of random integers $[\tilde{\rho}_n, [\tilde{\rho}_{n,m}]_{m=0,\ldots,1+D}]_{n=0,\ldots,1+D} \in_\$ (\mathbb{Z}_p)^{(3+D)\,(2+D)}$. It also selects a set of supplemental integers $[\chi_n]_{n=0,\ldots,1+D} \in_\$ (\mathbb{Z}_p)^{2+D}$ in a manner to be specified later. The simulator creates the decryption portion of the prototype private key for Id' as,

$$\left[\begin{array}{c} k_0, \\ [\,k_{n,(a)},\ k_{n,(b)}\,]_{n=0,\ldots,1+D} \end{array}\right] \leftarrow \left[\begin{array}{c} \hat{g}_2^{-\sum_{n=0}^{1+D} \chi_n \Theta_n / \bar{\Theta}_n} \displaystyle\prod_{n=0}^{1+D} \prod_{\ell=0}^{L'} (\hat{y}_{n,\ell}^{I_\ell})^{\tilde{\rho}_n}, \\ \left[\,\hat{g}_2^{\chi_n/\beta_n \bar{\Theta}_n}\, \hat{a}_n^{-\tilde{\rho}_n},\ \hat{g}_2^{\chi_n/\alpha_n \bar{\Theta}_n}\, \hat{b}_n^{-\tilde{\rho}_n}\,\right]_{n=0,\ldots,1+D} \end{array}\right],$$

and the re-randomization portion as, for all $m = 0, \ldots, 1 + D$,

$$\left[\begin{array}{c} f_{m,0}, \\ [\ f_{m,n,(a)},\ f_{m,n,(b)}\ ]_{n=0,\ldots,1+D} \end{array} \right] \leftarrow \left[\begin{array}{cc} \prod_{n=0}^{1+D} \prod_{\ell=0}^{L'} (\hat{y}_{n,\ell}^{I_\ell})^{\tilde{\rho}_{n,m}}, & \left[\begin{array}{cc} \hat{a}_n^{-\tilde{\rho}_{n,m}}, & \hat{b}_n^{-\tilde{\rho}_{n,m}} \end{array} \right]_{n=0,\ldots,1+D} \end{array} \right] ,$$

and also the delegation portion as, for all $\ell = 1 + L', \ldots, D$,

$$\left[\begin{array}{c} h_\ell, \\ [\ h_{m,\ell}\ ]_{m=0,\ldots,1+D} \end{array} \right] \leftarrow \left[\begin{array}{cc} \hat{g}_2^{-\sum_{n=0}^{1+D} \chi_n\, \theta_{n,\ell}/\bar{\Theta}_n} \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\tilde{\rho}_n}, & \left[\prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\tilde{\rho}_{n,m}} \right]_{m=0,\ldots,1+D} \end{array} \right] .$$

Once it has calculated the prototype key, the simulator feeds it to the regular *Derive* algorithm, and iteratively runs it using the sequence of identities $[I_0, \ldots, I_k]$ for $k = L' + 1, \ldots, L$. The end result is a private key for the requested identity; $\mathcal{B}$ gives it to $\mathcal{A}$ in response to the query.

According to Theorem 5, the simulator will issue a correctly distributed private key for Id provided that it manages to construct a prototype key for Id' that is itself correctly distributed. To prove the latter, we consider the change of variables, $\rho_n = \tilde{\rho}_n - z_2\, \chi_n/\alpha_n\, \beta_n\, \bar{\Theta}_n$, for $n = 0, \ldots, 1+D$. The new variables ρ_n are uniformly i.i.d. in $\mathbb{Z}_p$, but their values are unknown to $\mathcal{B}$ (as it lacks z_2). It is easy to see that under this substitution all $[k_{n,(a)}, k_{n,(b)}]$ always assume the same form as in the real scheme, $k_{n,(a)} = \hat{g}_2^{\chi_n/\beta_n\, \bar{\Theta}_n}\, \hat{a}_n^{-\tilde{\rho}_n} = \hat{g}_2^{\chi_n/\beta_n\, \bar{\Theta}_n}\, (\hat{g}^{\alpha_n})^{-z_2\, \chi_n/\alpha_n\, \beta_n\, \bar{\Theta}_n}\, \hat{a}_n^{-\rho_n} = \hat{a}_n^{-\rho_n}$, and also, $k_{n,(b)} = \hat{b}_n^{-\rho_n}$. As for $f_{m,0}$ and $[f_{m,n,(a)}, f_{m,n,(b)}]$, their expressions are unaffected by the change of variables, and are already in the correct form by construction. The same applies to the $h_{m,\ell}$. It remains to show that k_0 and the h_ℓ are well-formed, too. On the one hand, we have,

$$\begin{aligned} k_0 &= \hat{g}_2^{-\sum_{n=0}^{1+D} \chi_n\, \Theta_n/\bar{\Theta}_n} \prod_{n=0}^{1+D} \prod_{\ell=0}^{L'} (\hat{y}_{n,\ell}^{I_\ell})^{\tilde{\rho}_n} \\ &= \hat{g}_2^{-\sum_{n=0}^{1+D} \chi_n\, \Theta_n/\bar{\Theta}_n} \left(\prod_{n=0}^{1+D} \left((\hat{g}^{\Theta_n}\, \hat{g}_1^{\bar{\Theta}_n})^{\alpha_n\, \beta_n} \right)^{z_2\, \chi_n/\alpha_n\, \beta_n\, \bar{\Theta}_n} \right) \left(\prod_{n=0}^{1+D} \prod_{\ell=0}^{L'} (\hat{y}_{n,\ell}^{I_\ell})^{\rho_n} \right) \\ &= (\hat{g}^{z_1\, z_2})^{\sum_{n=0}^{1+D} \chi_n} \prod_{n=0}^{1+D} \prod_{\ell=0}^{L'} (\hat{y}_{n,\ell}^{I_\ell})^{\rho_n} . \end{aligned}$$

On the other hand, we have, for $\ell = 1 + L', \ldots, D$,

$$\begin{aligned} h_\ell &= \hat{g}_2^{-\sum_{n=0}^{1+D} \chi_n\, \theta_{n,\ell}/\bar{\Theta}_n} \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\tilde{\rho}_n} \\ &= \hat{g}_2^{-\sum_{n=0}^{1+D} \chi_n\, \theta_{n,\ell}/\bar{\Theta}_n} \left(\prod_{n=0}^{1+D} \left((\hat{g}^{\theta_{n,\ell}}\, \hat{g}_1^{\bar{\theta}_{n,\ell}})^{\alpha_n\, \beta_n} \right)^{z_2\, \chi_n/\alpha_n\, \beta_n\, \bar{\Theta}_n} \right) \left(\prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_n} \right) \\ &= (\hat{g}^{z_1\, z_2})^{\sum_{n=0}^{1+D} \chi_n\, \bar{\theta}_{n,\ell}/\bar{\Theta}_n} \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_n} . \end{aligned}$$

These values, k_0 and the h_ℓ, will assume the correct form provided that,

$$\sum_{n=0}^{1+D} \chi_n = 1 , \qquad \forall \ell \in \{1 + L', \ldots, D\} : \sum_{n=0}^{1+D} \chi_n\, \bar{\theta}_{n,\ell}/\bar{\Theta}_n = 0 ,$$

which constitutes a linear system of $1+D-L'$ equations of $2+D$ unknowns. It is easy to argue that this system admits a solution with overwhelming probability. The equations for $\ell \in \{1+L', \ldots, D\}$ together form an under-determined homogeneous linear sub-system, whose solutions fill a vectorial sub-space of dimension at least $(2+D)-(1+D-L') \geq 2$ in $(\mathbb{Z}_p)^{2+D}$, unless the sub-system is defective, which happens with probability at most $1/p$ since all the coefficients $\bar{\theta}_{n,\ell}/\bar{\Theta}_n$ are random. The outstanding equation is then readily satisfied. It remains to argue that setting the $[\chi_\cdot]$ in this manner leaves unimpaired the proper randomization of the key. This is clearly the case since a solution for $[\chi_\cdot]$ can be found prior to selecting any of the randomization exponents $[\tilde{\rho}_\cdot]$.

◇ *Challenge :*

When the adversary is ready to accept a challenge on the previously chosen target identity Id^*, it gives two message Msg_0 and Msg_1 to the simulator. The simulator selects a tuple of random integers $[r_n]_{n=0,\ldots,1+D} \in_\$ (\mathbb{Z}_p)^{1+D}$, picks a random bit $\delta \in_\$ \{0,1\}$, and outputs the challenge ciphertext,

$$\mathsf{CT}^* = \begin{bmatrix} E, \quad c_0, \\ [\, c_{n,(a)},\ c_{n,(b)} \,]_{n=0,\ldots,1+D} \end{bmatrix} \leftarrow \begin{bmatrix} \mathsf{Msg}_\delta \cdot Z^{-1}, \quad g_3, \\ \left[\, (g^{r_n})^{\beta_n \sum_{\ell=0}^{L^*} \theta_{n,\ell} I_\ell^*},\ (g_3\, g^{-r_n})^{\alpha_n \sum_{\ell=0}^{L^*} \theta_{n,\ell} I_\ell^*} \,\right]_{n=0,\ldots,1+D} \end{bmatrix} .$$

The challenge will have the same distribution as in a real attack whenever the Decision BDH tuple originally given to $\mathcal{B}$ was legitimate, *i.e.*, when $Z = \mathbf{e}(g_1, \hat{g}_2)^{z_3}$. To see this, pose $r = z_3$, note that $c_0 = g_3 = g^{z_3}$, and rewrite, for every $n = 0, \ldots, 1+D$,

$$c_{n,(a)} = (g^{r_n})^{\beta_n \sum_{\ell=0}^{L^*} \theta_{n,\ell} I_\ell^*} = \left(g^{\sum_{\ell=0}^{L^*} \theta_{n,\ell} I_\ell^*}\, g_1^0\right)^{\beta_n r_n} = \left(\prod_{\ell=0}^{L^*} (g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\beta_n I_\ell^*}\right)^{r_n} = \left(\prod_{\ell=0}^{L^*} (b_{n,\ell})^{I_\ell^*}\right)^{r_n} ,$$

$$c_{n,(b)} = (g^r\, g^{-r_n})^{\alpha_n \sum_{\ell=0}^{L^*} \theta_{n,\ell} I_\ell^*} = \left(\prod_{\ell=0}^{L^*} (g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\alpha_n I_\ell^*}\right)^{(r-r_n)} = \left(\prod_{\ell=0}^{L^*} (a_{n,\ell})^{I_\ell^*}\right)^{(r-r_n)} ,$$

thus exploiting the fact that $\sum_{\ell=0}^{L^*} \bar{\theta}_{n,\ell}\, I_\ell^* \pmod{p} = 0$ for the target identity. On the contrary, whenever Z is a random element of $\mathbb{G}_T$, which happens when $\mathcal{B}$ was given a bogus Decision BDH tuple, the challenge CT^* is statistically independent of δ in the view of the adversary.

◇ *Query :*

In the second probing phase, the adversary makes a number of additional private key queries on adaptively chosen identities distinct from Id^* and all its prefixes, exactly as in the first phase. The simulator responds as before.

◇ *Outcome :*

Eventually, the adversary emits a guess $\tilde{\delta} \in \{0,1\}$ as to which message the challenge ciphertext CT^* is an encryption of. The simulator forwards 1 if $\tilde{\delta} = \delta$ and 0 otherwise as its own guess regarding whether the input it initially received was a valid Decision BDH tuple.

This completes the simulation.

It is easy to see that the reduction works, since the simulation is perfect from beginning to end, unless the given BDH tuple was invalid, in which case the plaintext is independent of the challenge, as required. Since the reduction is time-efficient, and almost tight except for a tiny $(3+D)/p$ total probability of encountering an error condition upon each query, the theorem follows. □

C.2 Anonymity

The HIBE anonymity proof is based on the same type of simulation as for semantic security, except that now the reduction is from D-Linear instead of D-BDH. As in the IBE scheme of Section 4, we use "linear splittings" to conceal the identity in the ciphertext. We build a simulator that uses the given D-Linear tuple to perform such "splitting" for each identity component at a time. The complete proof is thus a hybrid argument; it consists of a sequence of games, where at each step, we show that the adversary cannot recognize a valid pair $[c_{n,(a)}, c_{n,(b)}]$ from a random pair $[\star, \star]$ in the challenge ciphertext, for each $n \in \{0, \ldots, 1 + D\}$.

Consider step 0 for the sake of illustration. The simulator must reduce the Decision Linear problem to the dilemma $[c_{0,(a)}, c_{0,(b)}]$ *vs.* $[\star, \star]$ presented to the adversary. To do so, the simulator will omit to choose secret exponents $[\alpha_0, \beta_0]$; instead, it will use the D-Linear instance to simulate the key extraction process without knowing the components $[\hat{y}_{0,\ell}]_{\ell=0,\ldots,D}$ in the master key.

We will face the same difficultly as in Section C.1 that responding to HIBE private key queries requires the simulation of not one but many interconnected secret randomization exponents. This, and the linear splitting, are all issues we have already encountered in earlier proofs. However, their combined appearance in this proof causes interactions that seriously complicate matters. For this reason, the formal proof of Theorem 7 will be fairly involved.

Combining Anonymity and Semantic Security. Since we have already established semantic security, all we need to show is a reduction in the restricted anonymity game in which the challenge ciphertext is a random message that is not given to the adversary.

We devise a hybrid experiment that consists of a sequence of games where the adversary is given progressively garbled challenges. At one end, the challenge ciphertext is genuine, exactly as in a real attack environment; at the other, it is random and thus independent of the identity. In the entire experiment, the adversary is given truthful public parameters and access to a private key oracle as in a real attack, so that the games in the sequence differ only in how the challenge ciphertexts are formed.

Let each instance of the symbol $\star$ denote an element sampled independently at random from the appropriate group. The challenges are then specified as follows:

$\mathsf{CT}^*_{\mathrm{real}} = [E, c_0, [c_{0,(a)}, c_{0,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]]$ — genuine ciphertext, as in a real attack;

$\mathsf{CT}^*_0 = [\star, c_0, [c_{0,(a)}, c_{0,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]]$ — ciphertext for a random message;

$\mathsf{CT}^*_1 = [\star, c_0, [\star, \star], [c_{1,(a)}, c_{1,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]]$ — first "linear pair" is random;

...

$\mathsf{CT}^*_n = [\star, c_0, [\star, \star], \ldots, [\star, \star], [c_{n,(a)}, c_{n,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]]$ — increasingly many corruptions;

...

$\mathsf{CT}^*_{(1+D)} = [\star, c_0, [\star, \star], \ldots, [\star, \star], [c_{1+D,(a)}, c_{1+D,(b)}]]$ — last remaining "linear pair";

$\mathsf{CT}^*_{(2+D)} = [\star, c_0, [\star, \star], \ldots, [\star, \star]]$ — all "linear pairs" replaced by random;

$\mathsf{CT}^*_{\mathrm{random}} = [\star, \star, [\star, \star], \ldots, [\star, \star]]$ — completely random ciphertext, *ipso facto* anonymous.

For each transition from one game to the next, we need to show that the adversary cannot tell the two games apart with non-negligible advantage. We already note the following:

- The last two games, $\mathsf{CT}^*_{(2+D)}$ and $\mathsf{CT}^*_{\text{random}}$, are exactly the same since the only outstanding component, $c_0 = g^r$, is random and independent of the entire attack and thus amounts to $\star$ (it is independent because there are no other components that depend on r).
- The very first transition, from $\mathsf{CT}^*_{\text{real}}$ to CT^*_0, corresponds exactly to the semantic security indistinguishability result we proved in the previous section, so we already know that the adversary cannot distinguish between them.

For these reasons, we only need to focus on the intermediate transitions. In all of them, the element E of the ciphertext is set to $\star$, which means that the simulator may completely disregard the challenge plaintext chosen by the adversary.

Proof of Theorem 7. It suffices to show that each of the middle $2 + D$ transitions (from CT^*_i to CT^*_{i+1} for $i = 0, \ldots, 1 + D$) is indistinguishable by the adversary. We show each of them to be indistinguishable under the Decision Linear assumption, in Lemmata 8 and 9. These results will establish the theorem. ☐

Lemma 8. *In the setting of Theorem 7, no adversary can distinguish Game #0 from Game #1, in time $\tilde{\tau} \approx \tau$, with advantage $\tilde{\epsilon} \approx \epsilon$, while making no more than q private key extraction queries.*

Proof. We reduce the Decision Linear problem in $\mathbf{G}$ to the adversary's task in the stated attack. We build a reduction algorithm, $\mathcal{B}$, that provides the adversary, $\mathcal{A}$, with a simulated attack environment. Algorithm $\mathcal{B}$ is given as input a Decision Linear problem instance consisting of a tuple $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, \hat{g}, \hat{g}^{z_1}, \hat{g}^{z_2}, Z] \in \mathbb{G}^5 \times \hat{\mathbb{G}}^3 \times \mathbb{G}$ for random exponents $[z_1, z_2, z_3, z_4] \in (\mathbb{Z}_p)^4$, where the test element $Z \in \mathbb{G}$ is either equal to $g^{z_3+z_4}$ or is a random element g^{z_5} for some $z_5 \in \mathbb{Z}_p$. For clarity, the problem instance supplied to $\mathcal{B}$ will be rewritten as,

$$[g, g_1, g_2, g_{31}, g_{42}, \hat{g}, \hat{g}_1, \hat{g}_2, Z] \in \mathbb{G}^5 \times \hat{\mathbb{G}}^3 \times \mathbb{G} .$$

The simulation is described as follows.

◇ *Open :*

The adversary $\mathcal{A}$ opens the game by announcing the identity $\mathsf{Id}^* = [I^*_0, I^*_1, \ldots, I^*_{L^*}]$ it wishes to attack, where $\mathcal{A}$ is allowed to choose the number of hierarchical components, $L^* \in \{1, \ldots, D\}$, although we impose that $I^*_0 = 1$ as usual.

◇ *Setup :*

To create public parameters, the simulator $\mathcal{B}$ starts by drawing a tuple of random non-zero integers $[\omega, [\alpha_n, \beta_n]_{n=1,\ldots,1+D}] \in_\$ (\mathbb{Z}_p^\times)^{3+2D}$, and a vector of random integers $[\theta_{0,\ell}]_{\ell=0,\ldots,D} \in_\$ (\mathbb{Z}_p)^{1+D}$. For each $n = 1, \ldots, 1+D$, it also selects a vector of pairs of integers $[\theta_{n,\ell}, \bar{\theta}_{n,\ell}]_{\ell=0,\ldots,D} \in_\$ (\mathbb{Z}_p)^{2(1+D)}$, subject to the constraint that $\sum_{\ell=0}^{L^*} \bar{\theta}_{n,\ell} I^*_\ell = 0 \pmod p$, where it is noted that the elements with indices greater than L^* are left unconstrained. Next, the simulator assigns,

$$\left[\begin{array}{c} \Omega,\ [\, a_{0,\ell},\ b_{0,\ell}\,]_{\ell=0,\ldots,D}, \\ [\, [\, a_{n,\ell},\ b_{n,\ell}\,]_{\ell=0,\ldots,D}\,]_{n=1,\ldots,1+D} \end{array}\right] \leftarrow \left[\begin{array}{c} \mathbf{e}(g,\hat{g})^\omega, \quad \left[\, g_1^{\theta_{0,\ell}},\ g_2^{\theta_{0,\ell}}\, \right]_{\ell=0,\ldots,D}, \\ \left[\, \left[\, (g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\alpha_n},\ (g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\beta_n}\, \right]_{\ell=0,\ldots,D}\, \right]_{n=1,\ldots,1+D} \end{array}\right] .$$

The adversary is provided with the public parameters, Pub, which include the context $\mathbf{G}$ and the elements Ω and $[[a_{n,\ell}, b_{n,\ell}]_{\ell=0,\ldots,D}]_{n=0,\ldots,1+D}$; their distribution is as in the real scheme.

To complete the setup, the simulator computes what it can of the private key. Note that the public parameter simulation pegs the exponents α_0 and β_0 from the real scheme to the respective unknowns z_1 and z_2 implicitly defined by the Decision Linear instance. $\mathcal{B}$ partially computes the master key, Msk, as, (except for the crossed-out vector of $\hat{y}_{0,\cdot}$.)

$$\begin{bmatrix} \hat{w},\ [\ \hat{a}_0,\ \hat{b}_0,\ \cancel{[\ \hat{y}_{0,\ell}\ ]_{\ell=0,\dots,D}}\ ], \\ [\ \hat{a}_n,\ \hat{b}_n,\ [\ \hat{y}_{n,\ell}\ ]_{\ell=0,\dots,D}\ ]_{n=1,\dots,1+D} \end{bmatrix} \leftarrow \begin{bmatrix} \hat{g}^{\omega},\quad [\ \hat{g}_1,\ \hat{g}_2,\ \cancel{[\ \hat{g}^{\alpha_0\,\beta_0\,\theta_{0,\ell}}\ ]_{\ell=0,\dots,D}}\ ], \\ \left[\ \hat{g}^{\alpha_n},\ \hat{g}^{\beta_n},\ [\ (\hat{g}^{\theta_{n,\ell}}\ \hat{g}_1^{\bar{\theta}_{n,\ell}})^{\alpha_n\,\beta_n}\ ]_{\ell=0,\dots,D}\ \right]_{n=1,\dots,1+D} \end{bmatrix}.$$

◇ *Query :*

In the first probing phase, the adversary makes a number of extraction queries on adaptively chosen identities distinct from Id^* and all its prefixes. Suppose that $\mathcal{A}$ makes such a query on $\mathsf{Id} = [I_0, \dots, I_L]$ where $I_0 = 1$. To prepare a response, $\mathcal{B}$ starts by determining the identity component of lowest index, L', such that $I_{L'} \neq I^*_{L'}$, letting $L' = L^* + 1$ in the event that Id^* is a prefix of Id. Under the stated rules of query, such an $L' \in \{1, \dots, D\}$ always exists and is uniquely defined in said interval. The private key is constructed in two steps. In the first step, $\mathcal{B}$ creates a private key for the identity $\mathsf{Id}' = [I_0, \dots, I'_L]$. Notice that Id' is either equal to or a prefix of Id, but not of Id^*. Define $\Theta_0 \leftarrow \sum_{\ell=0}^{L'} \theta_{0,\ell}\, I_\ell$. For $n = 1, \dots, 1+D$, also define $\Theta_n \leftarrow \sum_{\ell=0}^{L'} \theta_{n,\ell}\, I_\ell$ and $\bar{\Theta}_n \leftarrow \sum_{\ell=0}^{L'} \bar{\theta}_{n,\ell}\, I_\ell$, and note that all $\bar{\Theta}_n \neq 0 \pmod p$ except with negligible probability $\leq {}^{(1+D)}/_p$ over the choice of $[\bar{\theta}_{n,\ell}]$. To proceed, $\mathcal{B}$ picks a tuple of random integers $[\tilde{\rho}_0, [\tilde{\rho}_{0,m}]_{m=0,\dots,1+D}] \in_\$ (\mathbb{Z}_p)^{3+D}$, and, additionally, picks a random tuple $[\tilde{\rho}_n, [\tilde{\rho}_{n,m}]_{m=0,\dots,1+D}] \in_\$ (\mathbb{Z}_p)^{3+D}$ for every $n = 1, \dots, 1+D$. Moreover, $\mathcal{B}$ selects a supplemental collection of integers, $[\chi_n, [\chi_{n,m}]_{m=0,\dots,1+D}]_{n=1,\dots,1+D} \in (\mathbb{Z}_p)^{(3+D)\,(1+D)}$, subject to certain constraints to be discussed later. The simulator creates the decryption portion of the prototype private key for Id' as,

$$\begin{bmatrix} k_0,\ [\ k_{0,(a)},\ k_{0,(b)}\ ], \\ [\ k_{n,(a)},\ k_{n,(b)}\ ]_{n=1,\dots,1+D} \end{bmatrix} \leftarrow$$

$$\begin{bmatrix} \hat{w} \displaystyle\prod_{n=1}^{1+D} \left(\left(\hat{g}_2^{-\Theta_n/\bar{\Theta}_n} \right)^{\Theta_0\,\tilde{\rho}_0} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\tilde{\rho}_n} \right),\ \left[\ \hat{a}_0^{-\tilde{\rho}_0\,(1+D)},\ \hat{b}_0^{-\tilde{\rho}_0\,(1+D)}\ \right], \\ \left[\ \hat{a}_n^{-\tilde{\rho}_n}\ \hat{g}_2^{\chi_n\,\tilde{\rho}_0\,\Theta_0/\bar{\Theta}_n\,\beta_n},\ \hat{b}_n^{-\tilde{\rho}_n}\ \hat{g}_2^{\chi_n\,\tilde{\rho}_0\,\Theta_0/\bar{\Theta}_n\,\alpha_n}\ \right]_{n=1,\dots,1+D} \end{bmatrix},$$

and the re-randomization portion as, for all $m = 0, \dots, 1+D$,

$$\begin{bmatrix} f_{m,0},\ [\ f_{m,0,(a)},\ f_{m,0,(b)}\ ], \\ [\ f_{m,n,(a)},\ f_{m,n,(b)}\ ]_{n=1,\dots,1+D} \end{bmatrix} \leftarrow$$

$$\begin{bmatrix} \displaystyle\prod_{n=1}^{1+D} \left(\left(\hat{g}_2^{-\Theta_n/\bar{\Theta}_n} \right)^{\Theta_0\,\tilde{\rho}_{0,m}} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\tilde{\rho}_{n,m}} \right),\ \left[\ \hat{a}_0^{-\tilde{\rho}_{0,m}\,(1+D)},\ \hat{b}_0^{-\tilde{\rho}_{0,m}\,(1+D)}\ \right], \\ \left[\ \hat{a}_n^{-\tilde{\rho}_{n,m}}\ \hat{g}_2^{\chi_{n,m}\,\tilde{\rho}_{0,m}\,\Theta_0/\bar{\Theta}_n\,\beta_n},\ \hat{b}_n^{-\tilde{\rho}_{n,m}}\ \hat{g}_2^{\chi_{n,m}\,\tilde{\rho}_{0,m}\,\Theta_0/\bar{\Theta}_n\,\alpha_n}\ \right]_{n=1,\dots,1+D} \end{bmatrix},$$

and also the delegation portion as, for all $\ell = 1 + L', \dots, D$,

$$\begin{bmatrix} h_\ell, \\ [\ h_{m,\ell}\ ]_{m=0,\dots,1+D} \end{bmatrix} \leftarrow \begin{bmatrix} \displaystyle\prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_n},\ \left[\ \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_{n,m}}\ \right]_{m=0,\dots,1+D} \end{bmatrix}.$$

Once this is done, the second step for $\mathcal{B}$ is, starting with the prototype private key for Id' calculated above, to apply the *Derive* algorithm iteratively to obtain private keys for the sequence of identities $\mathsf{Id}_k = [I_0, \ldots, I_k]$ as k is incremented from $L'+1$ to L. The end result is a private key $\mathsf{Pvk}_{\mathsf{Id}}$ for the requested identity $\mathsf{Id} = [I_0, \ldots, I_L]$. The simulator $\mathcal{B}$ gives this key to $\mathcal{A}$ in response to the query.

According to Theorem 5, the returned key for Id will be correctly distributed whenever the key for Id' is. To see that the prototype key is indeed distributed correctly, we make the following change of variables, for all $n = 1, \ldots, 1+D$, and $m = 0, \ldots, 1+D$,

$$\rho_0 = \tilde{\rho}_0 \, (1+D) \ , \qquad \rho_{0,m} = \tilde{\rho}_{0,m} \, (1+D) \ ,$$
$$\rho_n = \tilde{\rho}_n - \chi_n \, \frac{z_2 \, \tilde{\rho}_0 \, \Theta_0}{\bar{\Theta}_n \, \alpha_n \, \beta_n} \ , \qquad \rho_{n,m} = \tilde{\rho}_{n,m} - \chi_{n,m} \, \frac{z_2 \, \tilde{\rho}_{0,m} \, \Theta_0}{\bar{\Theta}_n \, \alpha_n \, \beta_n} \ ,$$

which lets us rewrite the various components in their usual form. *In extenso*,

$$\begin{aligned}
k_0 &= \hat{w} \prod_{n=1}^{1+D} \left(\left(\hat{g}_2^{-\Theta_n \, \bar{\Theta}_n^{-1}} \right)^{\Theta_0 \, \tilde{\rho}_0} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\tilde{\rho}_n} \right) \\
&= \hat{w} \prod_{n=1}^{1+D} \left(\left(\hat{g}_2^{-\Theta_n \, \bar{\Theta}_n^{-1}} \right)^{\Theta_0 \, \tilde{\rho}_0} \left(\prod_{\ell=0}^{L'} (\hat{g}^{\theta_{n,\ell}} \, \hat{g}_1^{\bar{\theta}_{n,\ell}})^{I_\ell} \right)^{\chi_n \, z_2 \, \tilde{\rho}_0 \, \Theta_0 \, \bar{\Theta}_n^{-1}} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\rho_n} \right) \\
&= \hat{w} \prod_{n=1}^{1+D} \left(\left(\hat{g}_2^{-\Theta_n \, \bar{\Theta}_n^{-1}} \right)^{\Theta_0 \, \tilde{\rho}_0} \left(\hat{g}_2^{\Theta_n} \right)^{\chi_n \, \tilde{\rho}_0 \, \Theta_0 \, \bar{\Theta}_n^{-1}} \left(\hat{g}^{z_1 \, z_2 \, \bar{\Theta}_n} \right)^{\chi_n \, \tilde{\rho}_0 \, \Theta_0 \, \bar{\Theta}_n^{-1}} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\rho_n} \right) \\
&= \hat{w} \left(\hat{g}_2^{\tilde{\rho}_0 \, \Theta_0} \right)^{\sum_{n=1}^{1+D} (\chi_n - 1) \, \Theta_n \, \bar{\Theta}_n^{-1}} \prod_{n=1}^{1+D} \left((\hat{g}^{z_1 \, z_2})^{\chi_n \, \tilde{\rho}_0 \, \sum_{\ell=0}^{L'} \theta_{0,\ell} \, I_\ell} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\rho_n} \right) \\
&= \hat{w} \left(\hat{g}_2^{\tilde{\rho}_0 \, \Theta_0} \right)^{\sum_{n=1}^{1+D} (\chi_n - 1) \, \Theta_n \, \bar{\Theta}_n^{-1}} \prod_{n=1}^{1+D} \left(\left(\prod_{\ell=0}^{L'} (\hat{g}^{\alpha_0 \, \beta_0 \, \theta_{0,\ell}})^{I_\ell} \right)^{\chi_n \, \tilde{\rho}_0} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\rho_n} \right) \\
&= \hat{w} \left(\hat{g}_2^{\tilde{\rho}_0 \, \Theta_0} \right)^{\sum_{n=1}^{1+D} (\chi_n - 1) \, \Theta_n \, \bar{\Theta}_n^{-1}} \left(\prod_{\ell=0}^{L'} \hat{y}_{0,\ell}^{I_\ell \, \rho_0} \right)^{\sum_{n=1}^{1+D} \chi_n / (1+D)} \prod_{n=1}^{1+D} \prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell \, \rho_n} = \hat{w} \prod_{n=0}^{1+D} \prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell \, \rho_n} \ ,
\end{aligned}$$

where the last equation is predicated on the two following conditions,

$$\sum_{n=1}^{1+D} (\chi_n - 1) = 0 \ , \qquad \sum_{n=1}^{1+D} (\chi_n - 1) \, \Theta_n \, \bar{\Theta}_n^{-1} = 0 \ ; \qquad \text{(1a,1b)}$$

and as required, we find that $k_{0,(a)} = \hat{a}_0^{-\rho_0}$ and $k_{0,(b)} = \hat{b}_0^{-\rho_0}$; and also, for $n = 1, \ldots, 1+D$,

$$\begin{aligned}
k_{n,(a)} &= \hat{a}_n^{-\tilde{\rho}_n} \, \hat{g}_2^{\chi_n \, \tilde{\rho}_0 \, \Theta_0 / \bar{\Theta}_n \, \beta_n} = \hat{a}_n^{-\rho_n} \, \hat{g}^{-\alpha_n \, \chi_n \, z_2 \, \tilde{\rho}_0 \, \Theta_0 / \bar{\Theta}_n \, \alpha_n \, \beta_n} \, \hat{g}_2^{\chi_n \, \tilde{\rho}_0 \, \Theta_0 / \bar{\Theta}_n \, \beta_n} = \hat{a}_n^{-\rho_n} \ , \\
k_{n,(b)} &= \hat{b}_n^{-\tilde{\rho}_n} \, \hat{g}_2^{\chi_n \, \tilde{\rho}_0 \, \Theta_0 / \bar{\Theta}_n \, \alpha_n} = \hat{b}_n^{-\rho_n} \, \hat{g}^{-\beta_n \, \chi_n \, z_2 \, \tilde{\rho}_0 \, \Theta_0 / \bar{\Theta}_n \, \alpha_n \, \beta_n} \, \hat{g}_2^{\chi_n \, \tilde{\rho}_0 \, \Theta_0 / \bar{\Theta}_n \, \alpha_n} = \hat{b}_n^{-\rho_n} \ .
\end{aligned}$$

Using analogous calculations, we can derive a similar set of relations, such as, for $m = 0, \dots, 1+D$,

$$\begin{aligned}
f_{m,0} &= \prod_{n=1}^{1+D} \left(\left(\hat{g}_2^{-\Theta_n\, \bar{\Theta}_n^{-1}} \right)^{\Theta_0\, \tilde{\rho}_{0,m}} \left(\prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell} \right)^{\tilde{\rho}_{n,m}} \right) \\
&= \left(\hat{g}_2^{-\tilde{\rho}_{0,m}\, \Theta_0} \right)^{\sum_{n=1}^{1+D} (\chi_{n,m}-1)\, \Theta_n\, \bar{\Theta}_n^{-1}} \left(\prod_{\ell=0}^{L'} \hat{y}_{0,\ell}^{I_\ell\, \rho_{0,m}} \right)^{\sum_{n=1}^{1+D} \chi_{n,m}/(1+D)} \prod_{n=1}^{1+D} \prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell\, \rho_{n,m}} \\
&= \prod_{n=0}^{1+D} \prod_{\ell=0}^{L'} \hat{y}_{n,\ell}^{I_\ell\, \rho_{n,m}} \ ,
\end{aligned}$$

where for the last equality to hold we impose that, for all $m = 0, \dots, 1+D$,

$$\sum_{n=1}^{1+D} (\chi_{n,m} - 1) = 0 \ , \qquad \sum_{n=1}^{1+D} (\chi_{n,m} - 1)\, \Theta_n\, \bar{\Theta}_n^{-1} = 0 \ ; \tag{2a,2b}$$

in addition, we have the required $f_{m,0,(a)} = \hat{a}_0^{-\rho_{0,m}}$ and $f_{m,0,(b)} = \hat{b}_0^{-\rho_{0,m}}$; and furthermore, for all $n = 1, \dots, 1+D$,

$$\begin{aligned}
f_{m,n,(a)} &= \hat{a}_n^{-\tilde{\rho}_{n,m}}\, \hat{g}_2^{\chi_{n,m}\, \tilde{\rho}_{0,m}\, \Theta_0/\bar{\Theta}_n\, \beta_n} = \hat{a}_n^{-\rho_{n,m}}\, \hat{g}^{-\alpha_n\, \chi_{n,m}\, z_2\, \tilde{\rho}_{0,m}\, \Theta_0/\bar{\Theta}_n\, \alpha_n\, \beta_n}\, \hat{g}_2^{\chi_{n,m}\, \tilde{\rho}_{0,m}\, \Theta_0/\bar{\Theta}_n\, \beta_n} = \hat{a}_n^{-\rho_{n,m}} \ , \\
f_{m,n,(b)} &= \hat{b}_n^{-\tilde{\rho}_{n,m}}\, \hat{g}_2^{\chi_{n,m}\, \tilde{\rho}_{0,m}\, \Theta_0/\bar{\Theta}_n\, \alpha_n} = \hat{b}_n^{-\rho_{n,m}}\, \hat{g}^{-\beta_n\, \chi_n\, z_2\, \tilde{\rho}_{0,m}\, \Theta_0/\bar{\Theta}_n\, \alpha_n\, \beta_n}\, \hat{g}_2^{\chi_{n,m}\, \tilde{\rho}_{0,m}\, \Theta_0/\bar{\Theta}_n\, \alpha_n} = \hat{b}_n^{-\rho_{n,m}} \ .
\end{aligned}$$

As for the remaining components of the key, if, for each $\ell = 1+L', \dots, D$, and each $m = 0, \dots, 1+D$,

$$\sum_{n=1}^{1+D} \chi_n\, \frac{\bar{\theta}_{n,\ell}}{\bar{\Theta}_n}\, \frac{1}{\alpha_n\, \beta_n} = (1+D)\, \frac{\theta_{0,\ell}}{\Theta_0} \ , \qquad \sum_{n=1}^{1+D} \chi_{n,m}\, \frac{\bar{\theta}_{n,\ell}}{\bar{\Theta}_n}\, \frac{1}{\alpha_n\, \beta_n} = (1+D)\, \frac{\theta_{0,\ell}}{\Theta_0} \ , \tag{3,4}$$

then, we can equate, for every $\ell = 1+L', \dots, D$,

$$\begin{aligned}
h_\ell &= \left(\hat{g}^{z_1\, z_2\, \tilde{\rho}_0} \right)^{\overbrace{\left((1+D)\, \theta_{0,\ell} - \sum_{n=1}^{1+D} \chi_n\, \frac{\bar{\theta}_{n,\ell}}{\alpha_n\, \beta_n}\, (\Theta_0/\bar{\Theta}_n) \right)}^{0}} \left(\prod_{n=1}^{1+D} \hat{g}_2^{-\chi_n\, \theta_{n,\ell}\, \frac{\tilde{\rho}_0\, \Theta_0}{\bar{\Theta}_n\, \alpha_n\, \beta_n}} \right) \left(\prod_{n=1}^{1+D} \hat{y}_{n,\ell}^{\tilde{\rho}_n} \right) \\
&= \left(\hat{g}^{z_1\, z_2\, \theta_{0,\ell}\, \tilde{\rho}_0\, (1+D)} \right) \left(\prod_{n=1}^{1+D} \hat{g}^{-z_1\, z_2\, \chi_n\, \bar{\theta}_{n,\ell}\, \frac{\tilde{\rho}_0\, \Theta_0}{\bar{\Theta}_n\, \alpha_n\, \beta_n}} \right) \left(\prod_{n=1}^{1+D} \hat{g}_2^{-\chi_n\, \theta_{n,\ell}\, \frac{\tilde{\rho}_0\, \Theta_0}{\bar{\Theta}_n\, \alpha_n\, \beta_n}} \right) \left(\prod_{n=1}^{1+D} \hat{y}_{n,\ell}^{\tilde{\rho}_n} \right) \\
&= \left(\hat{g}^{z_1\, z_2\, \theta_{0,\ell}\, \rho_0} \right) \left(\prod_{n=1}^{1+D} (\hat{g}_1^{\bar{\theta}_{n,\ell}})^{-\chi_n\, \frac{z_2\, \tilde{\rho}_0\, \Theta_0}{\bar{\Theta}_n\, \alpha_n\, \beta_n}} \right) \left(\prod_{n=1}^{1+D} (\hat{g}_2^{\theta_{n,\ell}})^{-\chi_n\, \frac{\tilde{\rho}_0\, \Theta_0}{\bar{\Theta}_n\, \alpha_n\, \beta_n}} \right) \left(\prod_{n=1}^{1+D} \hat{y}_{n,\ell}^{\tilde{\rho}_n} \right) \\
&= \left(\hat{g}^{\alpha_0\, \beta_0\, \theta_{0,\ell}} \right)^{\rho_0} \left(\prod_{n=1}^{1+D} (g^{\theta_{n,\ell}}\, \hat{g}_1^{\bar{\theta}_{n,\ell}})^{-\chi_n\, \frac{z_2\, \tilde{\rho}_0\, \Theta_0}{\bar{\Theta}_n\, \alpha_n\, \beta_n}} \right) \left(\prod_{n=1}^{1+D} \hat{y}_{n,\ell}^{\tilde{\rho}_n} \right) \\
&= (\hat{y}_{0,\ell})^{\rho_0} \left(\prod_{n=1}^{1+D} (\hat{y}_{n,\ell})^{-\chi_n\, \frac{z_2\, \tilde{\rho}_0\, \Theta_0}{\bar{\Theta}_n\, \alpha_n\, \beta_n}} \right) \left(\prod_{n=1}^{1+D} \hat{y}_{n,\ell}^{\tilde{\rho}_n} \right) \\
&= (\hat{y}_{0,\ell})^{\rho_0} \prod_{n=1}^{1+D} (\hat{y}_{n,\ell})^{\rho_n} = \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_n} \ ,
\end{aligned}$$

and also, in the same way as above, for $\ell = 1 + L', \ldots, D$, and $m = 0, \ldots, 1 + D$,

$$h_{m,\ell} = \left(\prod_{n=1}^{1+D} \hat{g}_2^{\,-\chi_{n,m}\,\theta_{n,\ell}\,\frac{\tilde{\rho}_{0,m}\,\Theta_0}{\Theta_n\,\alpha_n\,\beta_n}} \right) \left(\prod_{n=1}^{1+D} \hat{y}_{n,\ell}^{\tilde{\rho}_{n,m}} \right) = \prod_{n=0}^{1+D} (\hat{y}_{n,\ell})^{\rho_{n,m}} \ .$$

To conclude this part of the argument, it suffices to show that the simulator can always choose a set of values for $[\chi_n, [\chi_{n,m}]_{m=0,\ldots,1+D}]_{n=1,\ldots,1+D} \in_\$ (\mathbb{Z}_p)^{(3+D)\,(1+D)}$ to satisfy all the constraints shown in (1–4), without jeopardizing the proper randomization of the prototype private key. As regards the latter point, observe that the constraints (1–4) are independent of the randomization exponents $[\tilde{\rho}_.]$, which means that the simulation can be freely randomized once the values for the $[\chi_.]$ have been determined.

To see that a solution for $[\chi_n]_{n=1,\ldots,1+D}$ always exists except with negligible probability, we first observe that these variables constitute a set of $1 + D$ unknowns in a linear system of $2 + D - L'$ equations given by (1a,1b,3). Since $2+D-L' \leq 1+D$, the system is never over-contrained. Next, we note that (1a) and (1b) together admit at least a solution in $(\mathbb{Z}_p)^{1+D}$ (*e.g.*, $\chi_1 = \ldots = \chi_{(1+D)} = 1$); these two equations therefore define an admissible affine sub-space $\mathbb{A}_1$ of dimension $D - 1$ in $(\mathbb{Z}_p)^{1+D}$. Then, remark that in (3), every unknown χ_n has an independent random coefficient in $\mathbb{Z}_p$ in every equation, so that (3) is a linear sub-system $\underline{\underline{A}}\,\underline{\chi} = \underline{b}$ where the matrix $\underline{\underline{A}}$ is random in $(\mathbb{Z}_p)^{(D-L')\times(1+D)}$; thus, unless $\underline{\underline{A}}$ is deficient, the sub-system defines a random affine sub-space $\mathbb{A}_2$ of dimension $(1 + D) - (D - L')$ in $(\mathbb{Z}_p)^{1+D}$. Observe that $\mathbf{P}(\mathbb{A}_1 \cap \mathbb{A}_2 = \emptyset) \leq {}^1\!/_p$. It follows that the entire system will be insoluble with negligible probability $\leq {}^2\!/_p$.

The same can be said to show that a solution for $[[\chi_{n,m}]_{m=0,\ldots,1+D}]_{n=1,\ldots,1+D}$ almost always exists, based on the fact that for each $m = 0, \ldots, 1 + D$, the constraints (2a,2b,4) form an independent linear system of $2 + D - L'$ equations of $1 + D$ unknowns exactly as above. Overall, we infer that the total probability that any of these $3 + D$ systems fails to admit a solution is $\leq {}^{(3+D)\,2}\!/_p$. Thus, taking into account the earlier failure probability $\leq {}^{(1+D)}\!/_p$, the simulator will be in good shape with probability $\geq 1 - {}^{(7+3\,D)}\!/_p$ upon answering this particular query.

⋄ *Challenge :*

When the adversary is ready to be challenged on the previously chosen target identity Id^*, the simulator selects a tuple of random integers $[[r_n]_{n=1,\ldots,1+D}] \in_\$ (\mathbb{Z}_p)^{1+D}$ and gives to the adversary the following challenge (where $\star$ is a random element of $\mathbb{G}_T$, *i.e.*, to encrypt a random message),

$$\mathsf{CT}^* = \begin{bmatrix} E, \ \ c_0, \ [\ c_{0,(a)}, \ c_{0,(b)}\ ], \\ [\ c_{n,(a)}, \ c_{n,(b)}\ ]_{n=1,\ldots,1+D} \end{bmatrix} \leftarrow \begin{bmatrix} \star, \ \ T, \ \ \left[\ (g_{42})^{\sum_{\ell=0}^{L^*} \theta_{0,\ell}\, I^*_\ell}, \ (g_{31})^{\sum_{\ell=0}^{L^*} \theta_{0,\ell}\, I^*_\ell}\ \right], \\ \left[\ (g^{r_n})^{\beta_n \sum_{\ell=0}^{L^*} \theta_{n,\ell}\, I^*_\ell}, \ (T\, g^{-r_n})^{\alpha_n \sum_{\ell=0}^{L^*} \theta_{n,\ell}\, I^*_\ell}\ \right]_{n=1,\ldots,1+D} \end{bmatrix} .$$

The challenge is well formed whenever $Z = g^{z_3+z_4}$. This can be seen by posing $r = z_3 + z_4$ and $r_0 = z_4$, under which substitutions the ciphertext can be rewritten as in the scheme. In particular,

$$c_{0,(a)} = g_2^{r_0 \sum_{\ell=0}^{L^*} \theta_{0,\ell}\, I^*_\ell} = \left(\prod_{\ell=0}^{L^*} (b_{0,\ell})^{I^*_\ell} \right)^{r_0} ,$$

$$c_{0,(b)} = g_1^{(r-r_0) \sum_{\ell=0}^{L^*} \theta_{0,\ell}\, I^*_\ell} = \left(\prod_{\ell=0}^{L^*} (a_{0,\ell})^{I^*_\ell} \right)^{r-r_0} ,$$

and furthermore, for every $n = 1, \ldots, 1+D$, since $0 = \sum_{\ell=0}^{L^*} \bar{\theta}_{n,\ell}\, I_\ell^* \pmod p$,

$$c_{n,(a)} = g^{\beta_n\, r_n\, \sum_{\ell=0}^{L^*} \theta_{n,\ell}\, I_\ell^*} = \left(g^{\sum_{\ell=0}^{L^*} \theta_{n,\ell}\, I_\ell^*}\, g_1^0\right)^{\beta_n\, r_n} = \left(\prod_{\ell=0}^{L^*} (g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\beta_n\, I_\ell^*}\right)^{r_n} = \left(\prod_{\ell=0}^{L^*} (b_{n,\ell})^{I_\ell^*}\right)^{r_n} ,$$

$$c_{n,(b)} = g^{\alpha_n\, (r-r_n)\, \sum_{\ell=0}^{L^*} \theta_{n,\ell}\, I_\ell^*} = \left(\prod_{\ell=0}^{L^*} (g^{\theta_{n,\ell}}\, g_1^{\bar{\theta}_{n,\ell}})^{\alpha_n\, I_\ell^*}\right)^{(r-r_n)} = \left(\prod_{\ell=0}^{L^*} (a_{n,\ell})^{I_\ell^*}\right)^{(r-r_n)} .$$

On the contrary, when $Z = g^{z_5}$ with $z_5 \in \mathbb{Z}_p$ random and independent, c_0 and all the pairs $[c_{n,(a)}, c_{n,(b)}]$ for $n \neq 0$ remain correctly jointly distributed, as can be shown by posing $r = z_5$ and arguing exactly as above; however, the two components $[c_{0,(a)}, c_{0,(b)}]$ are now statistically independent of Id^* in the view of the adversary. To see why, observe that a computationally unbounded adversary can uniquely determine r and all the r_n for $n \neq 0$ from the "good" part of the ciphertext, however there will be no r_0 that agrees with r, Id^*, and $[c_{0,(a)}, c_{0,(b)}]$. Indeed, since $g_{31} = g_1^{z_3}$ and $g_{42} = g_2^{z_4}$ where $[z_3, z_4]$ are independent of the rest of the simulation, it follows that the pair $[c_{0,(a)}, c_{0,(b)}]$ remains uniformly distributed in $(\mathbb{G})^2$ given Id^* and the rest of the ciphertext.

◇ *Query :*

In the second probing phase, the adversary makes a number of additional extraction queries on adaptively chosen identities distinct from Id^* and all its prefixes, as in the first phase. The simulator responds in the same manner.

◇ *Outcome :*

Eventually, the adversary emits a guess as to whether or not the challenge ciphertext CT^* was addressed to Id^*. The simulator forwards the adversary's output to its own challenger as its own guess as to whether the input it initially received was a valid Decision Linear tuple.

This concludes the description of the simulator.

The reduction is valid since the simulation is perfect from beginning to end, unless the given instance of the Decision Linear problem was an invalid tuple, in which case the first "linear pair" in the challenge ciphertext will be random. Specifically, the adversary is given a challenge that can be either,

$$[c_0, [c_{0,(a)}, c_{0,(b)}], [c_{1,(a)}, c_{1,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]] ,$$

or,

$$[c_0, [\star, \star], [c_{1,(a)}, c_{1,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]] ,$$

as required. The reduction is clearly time-efficient, and is tight except for a negligigle failure probability $\leq {}^{(7+3\,D)\,q}/_{p}$ for an attack that comprises q queries. Hence, the lemma follows. □

Lemma 9. *In the setting of Theorem 7, for each $n = 1, \ldots, 1+D$, no adversary can distinguish Game #n from Game #$n+1$, in time $\tilde{\tau} \approx \tau$, with advantage $\tilde{\epsilon} \approx \epsilon$, while making no more than q private key extraction queries.*

Proof. We can prove this lemma for each required transition almost exactly as the previous one, by exchanging the roles played by $[a_{0,\ell}, b_{0,\ell}, \hat{y}_{0,\ell}]$ with those played by $[a_{n,\ell}, b_{n,\ell}, \hat{y}_{n,\ell}]$ in the simulation, and taking care of the ramifications, *etc.* Specifically, α_0 and β_0 will now be chosen by the simulator, whereas the given instance of the Decision Linear problem will implicitly define $\alpha_n = z_1$ and $\beta_n = z_2$.

The other difference with the previous proof concerns the construction of the challenge CT^*. As expected, $[c_{n,(a)}, c_{n,(b)}]$ will be constructed using a technique analogous to the construction of $[c_{0,(a)}, c_{0,(b)}]$ in the previous proof, *i.e.*, based on g_{42} and g_{31} from the Decision Linear problem instance. However, we need no longer construct any of the pairs $[c_{n',(a)}, c_{n',(b)}]$ for $n' < n$, since the dilemma to be faced by the adversary is to distinguish between

$$[c_0, [\star, \star], \ldots, [\star, \star], [c_{n,(a)}, c_{n,(b)}], [c_{n+1,(a)}, c_{n+1,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]] \ ,$$

and

$$[c_0, [\star, \star], \ldots, [\star, \star], [\star, \star], [c_{n+1,(a)}, c_{n+1,(b)}], \ldots, [c_{1+D,(a)}, c_{1+D,(b)}]] \ .$$

We can simply set all the pairs that are random in both cases to randomly selected elements $[c_{n',(a)}, c_{n',(b)}] \in_{\$} (\mathbb{G})^2$, for $n' < n$. The rest of the proof is analogous to that of Lemma 8. $\square$