Electronic Colloquium on Computational Complexity, Report No. 167 (2012)

How powerful are the DDH hard groups?

Periklis A. Papakonstantinou[*] Charles W. Rackoff[†] Yevgeniy Vahlis[‡]

Abstract

The question whether Identity-Based Encryption (IBE) can be based on the Decisional Diffie-Hellman (DDH) assumption is one of the most prominent questions in Cryptography related to DDH. We study limitations on the use of the DDH assumption in cryptographic constructions, and show that it is impossible to construct a secure Identity-Based Encryption system using, in a black box way, only the DDH (or similar) assumption about a group. Our impossibility result is set in the generic groups model, where we describe an attack on any IBE construction that relies on oracle access to the group operation of randomly labelled group elements – a model that formalizes naturally DDH hardness.

The vast majority of existing separation results typically give separation from general primitives, whereas we separate a primitive from a class of number theoretic hardness assumptions. Accordingly, we face challenges in creating an attack algorithm that will work against constructions which leverage the underlying algebraic structure of the group. In fact, we know that this algebraic structure is powerful enough to provide generic constructions for several powerful primitives including oblivious transfer and chosen ciphertext secure public-key cryptosystems (note that an IBE generalizes such systems). Technically, we explore statistical properties of the group algebra associated with a DDH oracle, which can be of independent interest.

1 Introduction

In 1998 Dan Boneh [Bon98] surveyed the Decisional Diffie-Hellman problem; the survey begins with the proclamation:

"The Decision Diffie-Hellman assumption DDH is a gold mine."

Boneh's statement captured the excitement surrounding a recent spade of seminal works realized from the DDH assumption, including efficient pseudo-random functions [NR04] and efficient chosen-ciphertext encryption [CS98].

In the last fourteen years, the cryptography landscape has changed drastically. The emergence of pairing-based cryptography (and more recently lattices) has provided elegant and efficient constructions for several cryptosystems not even known to be feasible fourteen years ago. Some of the most prominent examples include Identity-Based Encryption (IBE) [BF01, Coc01, Sha84] and Homomorphic Encryption [Gen09].

While the excitement over this new frontier of results is well justified, we will explore if these new number theoretic techniques are actually required or whether we might be able to build such

[*]ITCS/IIIS, Tsinghua University, Beijing, PRC
[†]Dept. of Computer Science, University of Toronto, Toronto, ON, Canada
[‡]AT & T Security Research Center, New York City, NY, USA

ISSN 1433-8092

systems from the 1998 "gold mine" of DDH hard groups (e.g., groups of prime order with no efficiently computable bilinear map). Studying the feasibility of cryptography based on DDH hard groups is conceptually an important direction. First, solutions of a positive nature would give new candidate constructions under different hardness assumptions. It is possible (relative to the current state of knowledge) that there exist attacks on bilinear maps or lattice systems, but none against certain DDH groups[1]. Moreover, candidates for DDH hard groups have been scrutinized for relatively long periods of time, potentially increasing our confidence in them. Finally, there is a strong argument that knowing both the possibilities and limitations of DDH hard groups will deepen our fundamental understanding of the cryptographic systems in question. For instance, it is conceivable that a separation through a generic attack against a DDH-based system could provide new insights for existing systems.

We consider constructions of cryptographic primitives based on DDH hard groups that do not exploit any *non-generic* (or *non-black-box*) property of the group. Technically, we prove our results in the Generic Groups Model (GGM) of Shoup [Sho97], which has been criticized as too strong for proving *positive* results [KM07] (here we show a negative result). Our result rules out secure constructions for the whole class of generic constructions, which reflect almost all known uses of these types of DDH hard groups.

Our exploration on the limitations of DDH begins with Identity-Based Encryption (IBE), which is a public-key system where a user can encrypt to any other recipient knowing just the system's public parameters as well as a recipient's identity string. IBE is a natural case study due both to the fact that many of the more advanced crypto primitives imply its existence and the fact that there exist pairing-based [BF01], factoring-based [Coc01], and lattice-based [GPV08] solutions. Indeed, the recent success in building IBE schemes from a diverse set of number theoretic hardness assumptions might give us hope that IBE systems might be realizable from DDH hard groups.

Our Result We show that it is impossible[2] to securely build IBE generically from a DDH hard group; i.e. when we are given black-box access to the operation of a randomly labelled prime order group. One implication is that in some sense we rule out every IBE construction that doesn't take advantage of any special non-generic property of the group in question.

We prove our result in the generic group model (GGM). Shoup formalized the GGM as a model of computation where the group elements *are random* by a random embedding of $\mathbb{Z}_p := \mathbb{Z}/p\mathbb{Z}$, $p \in \mathbb{Z}$, into a finite set of strings; i.e. a group element is both a string and has algebraic meaning. The definition of this model is given in Section 2. Informally, an algorithm in this model makes queries to an oracle realizing the group operation, as induced by the embedding of $\mathbb{Z}_p$. An algorithm in the GGM, or *generic algorithm*, is computationally unbounded and the resource we measure is the number of oracle queries. Each query involves group elements that are either in the input or they appear as answers to previous queries. For example, if g, h are group elements (encoded as strings) listed on the input, one possible query is $g + h$; we assume that these are additive groups. The main point is that since the group element $(g + h)$ is a random string; intuitively, the only relation among group elements an attack algorithm can test is equality among group elements.

Several hardness assumptions, which intuitively rely on the fact that the presented group ele-

[1]Several bilinear map assumptions are actually strictly stronger than assuming the existence of DDH hard groups. For instance, Decisional Bilinear Diffie-Hellman relies on the target group of the bilinear map to be DDH hard.

[2]Our result can also be stated as it is commonly the case for fully black-box separations [RTV04]. Given the nature of our result, it seems more intuitive to state it in terms of oracle queries.

ments *look random*, can be shown to hold in the GGM. This includes the Decisional Diffie Hellman (DDH) assumption, which states that: there is a family of additive groups $\{G_n\}$ of prime order p_n that when presented in an appropriate encoding then the distribution of $(a \cdot g, b \cdot g, (ab) \cdot g)$ is computationally indistinguishable from $(a \cdot g, b \cdot g, c \cdot g)$ for randomly chosen $a, b, c \in \{0, \ldots, p_n - 1\}$, where p_n and the generator are also given to the distinguisher. Additionally, we show that in the GGM model we can break even a weak variant of semantic security of every IBE, which in particular separates in the black-box sense IBE from DDH.

A remark on black-box separations A black-box separation between two cryptographic primitives A and B is performed in a constructive way; loosely speaking, a so-called fully black-box separation is performed in an oracle (the black-box) setting, where what matters is the number of oracle queries. In this setting we show that A exists but B does not. Such a result can be understood as an impossibility result in reducing (in a black-box sense) the security of primitive B to A. It is worth noting that such impossibility results rule out only oracle (black-box) constructions. Often, such an oracle has a complex description fine tuned to allow one to break every B but still construct A. We stress that our separation result is with respect to a natural oracle that has been used, with the exception of [CDK$^+$12, DHT12], for positive results. In that sense, our work is similar in flavor to the original work of Impagliazzo and Rudich [IR89]. Furthermore, the Generic Groups Model is the model which precisely formalizes the intuition (i.e. the labels of group elements look random) about the hardness of DDH. As such, the techniques developed here can be of independent interest.

What's new? The main challenge in proving an impossibility result in the generic groups setting, as opposed to random oracles or trapdoor permutations, stems from the algebraic structure imposed on the random strings. In principle, an impossibility result in the GGM should employ new techniques, since, for example, in the GGM model Public Key Encryption exists (e.g. El-Gamal, Cramer-Shoup [CS98]), whereas there is no black-box PKE construction in the Random Oracle model [IR89]. Furthermore, although DDH yields a trapdoor permutation, there are DDH-based constructions where no such construction is known from generic trapdoors. For example, the Cramer-Shoup public key system ([GMM07] makes a step towards separating DDH from generic trapdoors), and in leakage resilient cryptography [ADVW12]. In principle, things are more involved than just what the above examples suggest. There are very subtle technical challenges raised by the algebraic structure of the oracle *in an IBE GGM-based system*, exhibited by examples that go beyond the purpose of this exposition. Here is the intuitive reason that makes an attack against an IBE system possible in the GGM. Take any IBE system that makes non-trivial use of its keys and start observing the keys associated with each identity. After a while there are *non-trivial algebraic relations* that necessarily hold between these secret keys. A key technical step of our contribution is to identify these relations as *linear relations* (or better: *affine*) over appropriate vector spaces. This tells us that:

> Generic use of DDH hardness relies on a "linear use of the group". Contrast this to bilinear pairings hardness that in practice make "non-linear use" of the group.

Our Attack Intuition We begin by giving an informal description of an IBE in the Generic Groups Model. Then, we sketch out a "first pass" attack algorithm. We put things in context by presenting this attack algorithm, which reflects some of our intuitive ideas of capturing linear relations in the secret key. Our eventual adversary will need to be *much* more complex and consists of multiple phases. Sketches about the reasons why the additional phases are needed are given in Sections 3 and 4.

Informally, an IBE system consists of four algorithms: *Setup, Key-generation, Encoding* and *Decoding* algorithm. All four algorithms have access to the same randomly chosen group oracle $\mathcal{O}$. Each algorithm's input and output is a string w together with a list of group elements $g_1, \ldots, g_m$. For consistency with the vector space notation all groups in this paper are additive. Each oracle query is a formal sum of the group elements in its input; e.g. $2g_1 + 3g_4$ (note that this is only with polynomial loss in the number of queries compared to Shoup's definition of the GGM). The oracle answers by valuating this to a group element. Setup creates the public parameters consisting of the master public key PP and the master private key MSK. Every algorithm (potentially) has access to PP, whereas only the Key-generation has access to MSK. The Key-generation issues for every identity ID a secret key $\mathrm{SK}_{\mathrm{ID}}$. Encryption is done using PP and ID, and decryption using $\mathrm{SK}_{\mathrm{ID}}$. We consider a weak form of semantic security (which makes an impossibility result stronger) where the adversary chooses a polynomial number of identities, for which he sees the secret keys, and a challenge identity ID^* distinct from the ones he chose before. Then, the adversary is presented with a ciphertext C_* to decrypt, where C_* is the encryption of a single bit. We say that the adversary wins if he decrypts correctly with polynomially many oracle queries and sufficient probability.

Our First Pass Attack: Suppose that each algorithm in the IBE system makes $\leq n$ queries. Choose $\mathrm{ID}_* = \mathrm{ID}_{n+1}$ and n identities $\mathrm{ID}_1, \ldots, \mathrm{ID}_n$, and get $\mathrm{SK}_{\mathrm{ID}_1}, \ldots, \mathrm{SK}_{\mathrm{ID}_n}$ and the encryption C_* of a random bit for ID_*. Then, for each identity the adversary encrypts and decrypts (say just once) a random bit, and it *records all the query-answer pairs* to the oracle. That is, a (query,answer) pair is of the form $(\sum_i a_i g_i, g)$. After this sampling phase construct a hybrid (random) group oracle $\mathcal{O}_{\mathsf{fake}}$ and an $\mathrm{MSK}_{\mathsf{fake}}$ consistent with (i) the recorded query-answer pairs, and (ii) PP, and simulate Setup and Key-generation to obtain $\mathsf{SK}_{\mathsf{ID}_*,\mathsf{fake}}$ and use them to decrypt C_*. By "consistent" we mean that it is consistent with respect to the actually constructed outputs, but also consistent with respect to any affine algebraic relations that can be deduced by the equalities between two distinct queries with the same oracle answer (see below for more details).

In the computation of a generic algorithm every newly presented group element is presented with its label, which by definition of the GGM is a random string. We wish to formalize the intuition that an adversary can only learn equality relations among distinct queries which evaluate to the same group elements. The main point is that at the same time this is roughly all algorithms of the IBE themselves can "learn" as well. For example, if at some point of the execution we have queries $\mathcal{O}(2g_1 + 3g_2) \mapsto u$ and $\mathcal{O}(g_1 + 5g_2) \mapsto u$ then an adversary that inspects this execution may infer that $2g_1 + 3g_2 - (g_1 + 5g_2) = g_1 - 2g_2$ corresponds to the identity of the group. Recall that the kernel of a linear map forms a vector space. In our case this vector space of *"equality vectors"* gives us a way[3] of expressing queries to a "fake oracle" as linear combinations of elements of the given group elements (one degenerate case is when we can find the discrete logarithms of all group elements g_i).

Although it doesn't work for all IBEs, our first pass attacker can break some natural candidate IBE systems. For instance, suppose that the master secret key consists of random exponents (recall

[3] Consider all possible rewritings (affine relations) of the formal sums in the queries given this vector space of equality relations. At a high level, this is all that both the adversary and the IBE algorithms can learn about the "syntax" of the exposed algebraic relations in the oracle. A fake oracle emulates answers of the actual oracle by creating a consistent hybrid and by making up "semantics" (discrete logarithms) which are correctly distributed.

though that the group is additive) $a_1, \ldots, a_n \in \mathbb{Z}_p$ for a group of order p and the public parameters consist of group elements $h_1, \ldots, h_n$ where $h_i = a_i \cdot g$, where g is a fixed generator which for simplicity we can fix to 1. Now, consider any publicly computable function $F(\mathrm{ID}) = c_{\mathrm{ID},1}, \ldots, c_{\mathrm{ID},n}$ where $c_{\mathrm{ID},i} \in \mathbb{Z}_p$. A natural IBE strategy would be to choose an F so that the secret key for user ID is of the form $\Sigma_i c_{\mathrm{ID},i} \cdot a_i$. To encrypt to a given identity one can form an ElGamal type ciphertext as $\big(sg, \Sigma_i (s \cdot c_{\mathrm{ID},i}) \cdot h_i\big)$.

For *any* such hash function F and whenever we have a set S of more than n identities, the secret key of one of these identities can be expressed as a linear combination of the secret keys of the other identities. While our attack is not geared to this exact system, it will naturally discover with non-negligible probability such relationships.

To obtain our full result that will attack any generic algorithm we will need to describe a significantly more complex attack algorithm that works against every IBE system. Given the conceptual complexity of the problem our attack consists of two independent steps. First, we reduce the security of the general IBE system to an IBE system with no group elements in the individual secret keys. Second, we perform an attack to the restricted system. Each of these steps is technically subtle.

Related Work In [IR89] Impagliazzo and Rudich introduced an approach for showing the impossibility of basing one cryptographic primitive on another in a black box manner. In that seminal paper they prove a separation between one-way permutations and secure key-agreement. Since then a large body of research has followed their basic methodology. We provide a survey of the most relevant results, and recommend reading [RTV04] for a more complete overview. Prior to our paper, there are two innovative works showing negative results in GGM: Dodis *et al* [DHT12] show the uninstantiability of the RSA Full Domain Hash signature scheme when the representation of the RSA is in the GGM, and Cramer *et al* [CDK$^+$12] show that a generalization of DDH is stronger than DDH. Part of our work is given in the PhD thesis [Pap10] containing the reduction presented in Section 3. Note that these results hold also for the more abstract/weaker model, where no explicit description of the group element is given, introduced by Maurer [Mau05]. In [BPR$^+$08], Boneh *et al* show an impossibility of secure black box constructions of IBE from trapdoor permutations. The techniques and subtleties that arise in [BPR$^+$08] are largely orthogonal to what we encounter in the current work due to the fact that trapdoor permutations have a very different algebraic structure. Our theorem is about GGM, it uses fundamentally new techniques (see "What's new?" above for a technical discussion); roughly speaking, the group structure allows to extend the "communication" among the parts of the IBE system in substantial ways that before weren't possible. Our novelty is in starting with a (generic model of) a number theoretic assumption; this provides several new challenges since we need to attack any construction that might try to leverage this algebraic structure. We also hope that our work will lead to other new explorations of the limits of DDH hard groups. For example, one interesting question is whether short signatures can be built in the standard model from such groups.

In [GKM$^+$00] Gertner *et al* show that public key encryption and Oblivious Transfer are incomparable under black-box reductions. They also show that trapdoor permutations cannot be constructed in a black-box way from public key encryption, or from trapdoor functions. In [GMR01] Gertner *et al* show that there is no black-box reduction from poly-to-one trapdoor functions to semantically secure public key encryption. Intuitively, [GMR01] shows that public key encryption is weaker than trapdoor functions because the latter allows the recovery of the complete input of the encryption algorithm, including the randomness. In [GGKT05] Gennaro *et al* show limits on the efficiency of cryptographic primitives constructed in a black-box way from basic tools such as one-way permutations, and trapdoor permutations. In [GMM07] Gertner *et al* prove that chosen ciphertext secure public key encryption cannot be constructed in a black-box way from semantically secure public key encryption (under some restrictions). In [Vah10] Vahlis showed new limitations on the uses of trapdoor permutations.

Of a slightly different flavor are the separations of [HHRS07, Sim98] that involve collision finding oracles, and [HH09] which is a very general impossibility result that establishes limitations of a very large class of assumptions.

The concept of identity-based encryption was proposed by Shamir [Sha84]. Several years later Boneh and Franklin [BF01, BF03] and Cocks [Coc01] independently gave two different constructions respectively based on bilinear groups and the quadratic residuosity problem, both in the random oracle model. Following their work different standard model constructions of IBE systems were given [BB04a, BB04b, CHK03, Gen06, Wat05]. More recently multiple lattice-based constructions have emerged [ABB10a, ABB10b, CHKP10, GPV08]. Altogether there are IBE constructions based on factoring-type assumptions, bilinear groups, and lattices, but nothing related to DDH hard groups.

2 Preliminaries

Linear algebra We shall assume familiarity with basic concepts from linear algebra (see e.g. [Lan87]). We denote matrices by capital bold letters, e.g. $\mathbf{A}$, and vectors by lower case bold letters, e.g. $\mathbf{v}$. Every group is in additive notation; i.e. if $\mathbb{G}$ is a group of order p, then for $g \in \mathbb{G}$, $c \in \mathbb{Z}_p$ we write cg (rather than g^c), where $cg = \underbrace{g + \cdots + g}_{c \text{ times}}$. In our constructions and analyses we often consider formal sums of group elements with coefficients in $\mathbb{Z}_p$, i.e. vectors in $\mathbb{Z}_p[G]$. For a formal sum $\mathbf{a} \in \mathbb{Z}_p[\mathbb{G}]$ with zero coefficients outside $g = \{g_1, \ldots, g_n\}$, $t = \{t_1, \ldots, t_m\} \subseteq \mathbb{G}$ we write $\mathbf{a} = \mathbf{a}^{(\mathbf{g})} + \mathbf{a}^{(\mathbf{t})}$ to denote separately its g_i's and t_i's component. For $R \subseteq \mathbb{Z}_p[\mathbb{G}]$ and $X \subseteq \mathbb{G}$, $R^{(\mathbf{X})} \stackrel{\text{def}}{=} \{\mathbf{q}^{(\mathbf{X})} \mid \mathbf{q} \in R\}$.

The Generic Groups Model The Generic Groups Model provides an algorithm with oracle access to a "perfectly secure" finite cyclic group of order N, by encoding each group element, through an *encoding function*, as a random string of length $n \approx \log N$. The following definition is equivalent, up to a polynomial number of queries, to Shoup's [Sho97] original definition.

Definition 1 (Generic Groups Model)**.** *Let $p \in \mathbb{Z}^+$ be a prime. Let $S \subseteq \{0,1\}^{O(\log p)}$ such that $|S| \geq p$. Let $\sigma : \mathbb{Z}_p \hookrightarrow S$ be an injection, and let $\mathbb{G}$ be the group induced on* $\mathrm{Im}(\sigma)$ *by addition in $\mathbb{Z}_p$. The function σ is called an* encoding function *for $\mathbb{Z}_p$. $\mathbb{Z}_p[\mathbb{G}]$ denotes the set of formal sums with coefficients from $\mathbb{Z}_p$ over $\mathbb{G}$. We define $\mathcal{O} : \mathbb{Z}_p[\mathbb{G}] \to \mathbb{G}$, where $\mathcal{O}(\alpha_1 g_1 + \cdots + \alpha_p g_p) = \alpha_1 g_1 +_{\mathbb{G}} \cdots +_{\mathbb{G}} \alpha_p g_p$, where the RHS denotes the evaluation of the formal sum to a group element. When there is no ambiguity we write $+$ instead of $+_{\mathbb{G}}$.*

A generic algorithm *A is a probabilistic algorithm (or with randomness in its input) that takes inputs and produces outputs of the form $(w, g_1, \ldots, g_k) \in (\{0,1\}^* \times \mathbb{G}^k)$ for an arbitrary $k \in \mathbb{N}$. A is given access to $\mathcal{O}$* restricted to sums that have non-zero coefficients only for the elements $g_1, \ldots, g_k$.

To allow meaningful computations every generic algorithm is given in its input a generator $\mathbf{1}$*, where for simplicity* $\mathbf{1} \overset{\text{def}}{=} \sigma(1)$*.*

Note that generic algorithms can treat $g \in \mathbb{G}$ both as a string and as a group element. In fact, this factors in the technical complication of our attack.

Additional Notation We now describe several notational conventions that we adopt throughout the paper.

Query transcripts. For a generic algorithm A we use the notation T_A (the input to A will be clear from context) to denote the set of pairs $(\mathbf{q}, g) \in \mathbb{Z}_p[\mathbb{G}] \times \mathbb{G}$, also denoted as $\mathbf{q} \mapsto g$, where $\mathbf{q}$ is a query that was asked during the computation of A, and $g = \mathcal{O}(\mathbf{q})$.

Discrete logarithms. Given an encoding function σ, an oracle query $\mathbf{q} \in \mathbb{Z}_p[\mathbb{G}]$ has a corresponding discrete logarithm value under σ: if $\mathbf{q} = a_1 g_1 + \cdots + a_p g_p$ then we denote $\text{dlog}_\sigma(\mathbf{q}) \overset{\text{def}}{=} a_1\sigma^{-1}(g_1) + \cdots + a_p\sigma^{-1}(g_p)$ with addition over $\mathbb{Z}_p$, and omit σ from the subscript when there is no ambiguity. Given a set $Q \subseteq \mathbb{Z}_p[\mathbb{G}]$ we extend the notation $\text{dlog}_\sigma(Q) \overset{\text{def}}{=} \bigcup_{\mathbf{q}\in Q} \{\text{dlog}_\sigma(\mathbf{q})\}$. Similarly, for a set T of query-answer pairs, we define $\text{dlog}_\sigma(T) \overset{\text{def}}{=} \bigcup_{(\mathbf{q},g)\in T} \{\text{dlog}_\sigma(\mathbf{q})\}$.

Partial encoding functions. Let T be a set of query-answer pairs. We say that T defines a partial encoding function σ_T if σ_T is the function of the smallest domain for which $\sigma_T(\mathbf{q})$ is defined for all mappings $\{\mathbf{q} \mapsto g\} \in T$, where there exists a unique set $X \subseteq \mathbb{Z}_p$ such that for all encoding functions σ, $\text{dlog}_\sigma(T) = X$, and $\sigma(1) = \mathbf{1}$. We also say that $\text{dlog}_{\sigma_T}(\mathbf{q})$ is well-defined if for every $g \in S$ which appears in $\mathbf{q}$ with a non-zero coefficient, $\sigma_T^{-1}(g)$ is defined.

Basis replacement. Given a query $\mathbf{q} \in \mathbb{Z}_p[\mathbb{G}]$, and a set of mappings (query-answer pairs) T, we write $\mathsf{CHB}_T(\mathbf{q})$ to denote the following iterative transformation: (i) initially $\mathbf{q}' \leftarrow \mathbf{q}$, (ii) repeat the following for each $\{\mathbf{u} \mapsto g\} \in T$: let a_g be the coefficient of g' in the formal sum $\mathbf{q}$; set $\mathbf{q}' \leftarrow \mathbf{q}' - a_g g + a_g \mathbf{u}$.

Observe that oracle valuations are stable/invariant and independent of the order of rewritings, under this change of base operation, when T contains query transcripts from the same oracle. Intuitively, this operation makes a "best effort" to express a query made over group elements on which we do not have oracle access to group elements where we currently do have oracle access.

Vector space of zero sums. Given a set T of query-answer pairs, we define $\mathsf{Eq}(T)$ to be the set of all equality relations between group elements that can be derived from T. Specifically,

$$\mathsf{Eq}(T) \overset{\text{def}}{=} \text{span}\left\{\mathbf{q} - \mathbf{q}' \middle| \exists g \in \mathbb{G} \text{ s.t. } \mathbf{q} \mapsto g, \mathbf{q}' \mapsto g \in T\right\}$$

Identity Based Encryption An IBE system consists of four probabilistic algorithms ($\mathit{Setup}, \mathit{KeyGen}, \mathit{Enc}, \mathit{Dec}$). In our case an IBE system also specifies a set $\mathcal{ID}$ of valid identities, and $p, m, n \in \mathbb{N}$, where p is a prime and $\varepsilon \in (1/2, 1]$ as part of its description, and the algorithms are generic with group size p, where p is at most exponentially bigger than n. Each IBE algorithm makes at most m queries in a single execution, n determines the number of random bits and output lengths, and ε determines the level of correctness; i.e. a correctly encrypted ciphertext is decrypted correctly with probability $\geq \varepsilon$.

The functionality of each algorithm is as follows: algorithm $\mathit{Setup}(\text{MSK})$ takes as input a master secret key $\text{MSK} \in \{0,1\}^n$; it outputs public parameters $\text{PP} = (w_{pp}, g_1, \ldots, g_m)$ where $w_{pp} \in \{0,1\}^*$

and $g_1, \dots, g_m \in \mathbb{G}$. Algorithm $\mathit{KeyGen}(\mathrm{MSK}, \mathrm{ID})$ is deterministic, and it takes as input the master secret key MSK, and an identity $\mathrm{ID} \in \mathcal{ID}$. It outputs the private key $\mathrm{SK_{ID}} = (w_{\mathrm{ID}}, z_1, \dots, z_m)$ for identity ID. The encryption algorithm $\mathit{Enc}(\mathrm{PP}, \mathrm{ID}, b; r)$ encrypts a bit b for a given identity ID using public parameters PP and randomness $r \in \{0,1\}^n$; it outputs a ciphertext $C = (w_c, t_1, \dots, t_m)$. The decryption algorithm $\mathit{Dec}(\mathrm{SK_{ID}}, C)$ decrypts a ciphertext C using the private key $\mathrm{SK_{ID}}$ and is deterministic. Note that making the algorithm K deterministic does not restrict the type of IBE constructions that our results cover since any secure IBE with a randomized K can be converted into a secure IBE with a deterministic K by applying a pseudo-random function to the identity in order to obtain (pseudo) random bits for K.

An IBE system must satisfy the following correctness property, for correctness parameter ϵ: for every $\mathrm{ID} \in \{0,1\}^n$, $b \in \{0,1\}$

$$\Pr\left[\begin{array}{l} \mathrm{PP} \leftarrow \mathit{Setup}(\mathrm{MSK}), \\ \mathrm{SK_{ID}} \leftarrow \mathit{KeyGen}(\mathrm{MSK}, \mathrm{ID}), \\ C \leftarrow_{\mathrm{R}} \mathit{Enc}(\mathrm{PP}, \mathrm{ID}, b) \end{array} ; \ \mathit{Dec}(\mathrm{SK_{ID}}, C) = b\right] \geq \varepsilon$$

Definition of Security For security we use a weakened version of the IBE semantic security, which is defined in [BF03]. The weak semantic security is defined using a game in which the attacker starts by choosing a polynomially long (in n) sequence of identities for which it wishes to obtain private keys, along with the challenge identity that it is going to try and break. This is done even before the adversary is given the public parameters. The adversary is then given the public parameters, a sequence of private keys for the above identities (except the challenge identity), and an encryption of a random bit for the challenge identity using the public parameters. More precisely, weak semantic IBE security is defined using a game between a challenger and an adversary $\mathcal{A}$ in which both parties are given a security parameter λ.

Setup: The adversary submits a challenge identity ID_*, and a sequence of identities $\mathrm{ID}_1, \dots, \mathrm{ID}_l$ where l where $l(\cdot)$ is some polynomial.

Private Keys: The challenger chooses at random $\mathrm{MSK} \in \{0,1\}^n$. It then computes $\mathrm{PP} \leftarrow \mathit{Setup}(\mathrm{MSK})$, and $\mathrm{SK}_i \leftarrow \mathit{KeyGen}(1^\lambda, \mathrm{MSK}, \mathrm{ID}_i)$ for $1 \leq i \leq l$. The tuple $(\mathrm{PP}, \mathrm{SK}_1, \dots, \mathrm{SK}_l)$ is given to the adversary.

Challenge: The challenger chooses at random $r \in \{0,1\}^n$, $b \in \{0,1\}$. It then computes $C_* \leftarrow \mathit{Enc}(\mathrm{PP}, \mathrm{ID}, b; r)$, and C_* is given to the adversary.

Guess: The adversary outputs a guess $b' \in \{0,1\}$, and wins if $b' = b$.

We define the advantage of the adversary $\mathcal{A}$ in attacking the scheme $\mathcal{E}$ as $\mathrm{Adv}_{\mathcal{E},\mathcal{A}} \stackrel{\mathrm{def}}{=} \left|\Pr[b = b'] - \frac{1}{2}\right|$. The probability is over the random bits used by the challenger and the adversary.

3 Eliminating Group Elements From Individual Private Keys

The impossibility is shown in two steps. Here we present the first step, where we show that every IBE in the Generic Groups Model can be converted to one that does not have any group elements in the private keys issued to the individual identities. This happens with a subconstant loss in correctness. In Section 4 we show how to attack a not-perfectly-correct[4] IBE restricted not to have group elements in the private keys.

Theorem 1. *Let $\mathcal{IBE}$ be an IBE with parameters (n, m, ε), then $\mathcal{RIBE}$ obtained from $\mathcal{IBE}$ through the transformation in Section 3.1 is an IBE with parameters $(n,' m', \varepsilon - 1/\mathsf{poly}(n))$, where n' and m' are polynomials in n and m respectively.*

Converting the key generation algorithm Let us consider the significance of group elements in individual private keys. Let us denote by $g_1, \ldots, g_m$ the group elements in the master public key, by $t_1, \ldots, t_m$ the group elements in a ciphertext that is encrypted to an identity ID, and by $z_1, \ldots, z_m$ the elements in a secret key $\mathrm{SK}_{\mathrm{ID}}$ of identity ID. Now, the encryption and decryption algorithms can use their oracles to generate formal sums of the group elements $g_1, \ldots, g_m$ and $z_1, \ldots, z_m$ respectively. As we already discussed in the introduction, intuitively the only queries that are useful for security are pairs of queries $(\mathbf{q}, \hat{\mathbf{q}})$ where $\mathbf{q}$ is asked by the encryption algorithm, $\hat{\mathbf{q}}$ is asked by the decryption algorithm, and $\mathcal{O}(\mathbf{q}) = \mathcal{O}(\hat{\mathbf{q}})$. If we break up each query into the components that correspond to the different sets of group elements (constant term, g_i's, t_i's, and z_i's):

$$\mathbf{q} = \mathbf{q}^{(\mathbf{1})} + \mathbf{q}^{(\mathsf{g})} \quad \text{and} \quad \hat{\mathbf{q}} = \hat{\mathbf{q}}^{(\mathbf{1})} + \hat{\mathbf{q}}^{(\mathsf{t})} + \hat{\mathbf{q}}^{(\mathsf{z})}$$

we obtain the relation $\hat{\mathbf{q}}^{(\mathsf{z})} = \mathbf{q}^{(\mathbf{1})} - \hat{\mathbf{q}}^{(\mathbf{1})} + \mathbf{q}^{(\mathsf{g})} - \hat{\mathbf{q}}^{(\mathsf{t})}$, where $\mathbf{q}^{(\mathbf{1})} = a\mathbf{1}$ for some $a \in \mathbb{Z}_p$. We would like to express each group element z_i in the secret key $\mathrm{SK}_{\mathrm{ID}}$ as a formal sum of other group elements that are available to the decryption. Let us examine what these elements are: the decryption algorithm has access, in addition to the group elements in $\mathrm{SK}_{\mathrm{ID}}$, to the group elements t_i in ciphertext that is being decrypted. Without loss of generality assume that *Dec* has access to the group elements g_i from the public key. We can now attempt to convert any *KeyGen* algorithm that produces keys that contain group elements into a new algorithm *KeyGen*$'$ that outputs only a string: *KeyGen*$'$ will first simulate *KeyGen* to obtain $\mathrm{SK}_{\mathrm{ID}}$ (which may contain group elements). It will then encrypt and decrypt a random bit many times, using $\mathrm{SK}_{\mathrm{ID}}$ for decryption, and observe which relations of the above form appear.

The above relation seemingly allows us to express formal sums of z_i as a combination of g_i's and t_i's. The trouble is that unlike the g_i's that are fixed in the global public key, the group elements t_i in the ciphertext can be different during each decryption. Therefore, the more encryptions we do the more the t_i's. Also, learning relations between z_i's and the t_i's for one ciphertext can be completely useless for decrypting others. In the above (oversimplified) description, *KeyGen*$'$ generated the ciphertexts to be decrypted on its own, and can observe the oracle queries that are made by *Enc*. Consequently, for each group element t_i in the ciphertext, *KeyGen*$'$ observes at least one query $\mathbf{q}_i$ of the form $\mathbf{q}_i = \mathbf{q}_i^{(\mathbf{1})} + \mathbf{q}_i^{(\mathsf{g})}$ such that $\mathcal{O}(\mathbf{q}_i) = t_i$. This allows us to express each t_i as a formal sum of g_i's, which in turn allows us to convert any relation of the form $\hat{\mathbf{q}}^{(\mathsf{z})} = \mathbf{q}^{(\mathbf{1})} - \hat{\mathbf{q}}^{(\mathbf{1})} + \mathbf{q}^{(\mathsf{g})} - \hat{\mathbf{q}}^{(\mathsf{t})}$ to one that describes $\hat{\mathbf{q}}^{(\mathsf{z})}$ as a formal sum of g_i's only by replacing $\hat{\mathbf{q}}^{(\mathsf{t})}$ with the appropriate formal sum of g_i's.

Let us now go back to our modified key generation algorithm *KeyGen*$'$. Let T be the set of all formal sums $\mathbf{q}^{(\mathsf{z})}$ for which we learn an equal formal sum of g_i's during the repeated encryption-decryption process. Now, consider a new formal sum $\hat{\mathbf{q}}^{(\mathsf{z})}$. If $\hat{\mathbf{q}}^{(\mathsf{z})} \in \mathsf{Eq}(T)$ then we already know

[4]Note that even if the original IBE is perfectly correct, since perfect correctness is lost in the transformation our attack in Section 4 should necessarily work against ϵ-correct IBEs.

how to express $\hat{\mathbf{q}}^{(\mathsf{z})}$ as a formal sum of g_i's. On the other hand, the event $\hat{\mathbf{q}}^{(\mathsf{z})} \notin \mathsf{Eq}(T)$ can only occur m times, where m is the number of group elements in the key $\mathrm{SK}_{\mathrm{ID}}$ (and therefore the dimension of $\mathsf{Eq}(T)$). Therefore, relying on the dimension of $\mathsf{Eq}(T)$ (which is exponentially smaller than the order of the group), and by sampling the number of repetitions of the simulated encryption-decryption from m^c times we obtain that in the "actual" encryption and decryption the probability that a linearly independent $\hat{\mathbf{q}}^{(\mathsf{z})}$ appears is at most $1/m^{c-1}$. Therefore, after this simulation the new key-generation algorithm outputs T instead of the group elements, in the hope that all algebraic relations useful to the decryption algorithm are already there. That is, now the secret key of ID is now *just a string* containing the description of the set T.

The modified decryption algorithm The following is a high level overview of the decryption algorithm, which is the most involved part of the transformation. To decrypt using just the table T instead of the group elements z_i, we change the decryption algorithm as follows. The new decryption algorithm simulates *Dec*, and let $\mathbf{q} = \mathbf{q}^{(\mathbf{1})} + \mathbf{q}^{(\mathsf{z})} + \mathbf{q}^{(\mathsf{t})}$ be a query. If $\mathbf{q}^{(\mathsf{z})} \in \mathsf{Eq}(T)^{(\mathsf{z})}$ then using elementary linear algebra we can syntactically replace the z_i's and construct a semantically equivalent (i.e. same discrete logarithm) one, $\hat{\mathbf{q}} = \hat{\mathbf{q}}^{(\mathbf{1})} + \hat{\mathbf{q}}^{(\mathsf{g})} + \mathbf{q}^{(\mathsf{t})}$. The difficulty occurs when $\mathbf{q}^{(\mathsf{z})} \notin \mathsf{Eq}(T)^{(\mathsf{z})}$. One thing we could try to do is to respond with a random element from S. This works only when there is no query $\mathbf{q}'$ that was asked before and such that $\mathcal{O}(\mathbf{q}) = \mathcal{O}(\mathbf{q}')$ (there are interesting examples showing that such $\mathbf{q}'$ can exist). Therefore, before responding with a random element we make sure that there is no equivalent[5] query that can already be inferred from T. Dealing with this notion of equivalence is the bulk of the transformation and its analysis.

3.1 The Transformation

Let $\mathcal{IBE} = \langle \mathit{Setup}, \mathit{KeyGen}, \mathit{Enc}, \mathit{Dec} \rangle$ be an IBE in the generic group model with parameters p, m, n, ε. Let $c \in \mathbb{N}$ be a constant to be determined later. We construct an IBE $\mathcal{RIBE} = \langle \mathit{Setup}', \mathit{KeyGen}', \mathit{Enc}', \mathit{Dec}' \rangle$ as follows:

***Setup*′(MSK):** Run $\mathrm{PP} = (w_{\mathrm{PP}}, g_1, \ldots, g_m) \leftarrow \mathit{Setup}(\mathrm{MSK})$ and output PP.

***KeyGen*′(MSK, ID):** Run
$\mathrm{SK}_{\mathrm{ID}} = (w_{\mathrm{ID}}, z_1, \ldots, z_m) \leftarrow \mathit{KeyGen}(\mathrm{MSK}, \mathrm{ID})$. For sufficiently large constant $c > 0$, randomly choose $k \in \{n^c, \ldots, 2n^c\}$ and construct a table $T \in (\mathbb{Z}_p[\mathbb{G}] \times \mathbb{G})^{kn}$ mapping queries to answers whose entries are from the following procedure. Repeat the following k times:

1. Choose a random $b \in_{\mathrm{R}} \{0, 1\}$ and compute
$(C, T_{\mathit{Enc}}) \leftarrow_{\mathrm{R}} \mathit{Enc}(\mathrm{PP}, \mathrm{ID}, b)$ where $C = (w_c, t_1, \ldots, t_m)$. Record in T all query-answer pairs asked by *Enc* by setting $T \leftarrow T \cup T_{\mathit{Enc}}$. Note that T_{Enc} contains only mappings of the form $\{a_0\mathbf{1} + a_1 g_1 + \cdots + a_m g_m \mapsto t\}$, the coefficients of all elements except $\mathbf{1}, g_1, \ldots, g_m$ are zero.
2. Run $\mathit{Dec}(\mathrm{SK}_{\mathrm{ID}}, C)$, and let $\mathbf{q}_{\mathsf{dec},1}, \ldots, \mathbf{q}_{\mathsf{dec},n}$ be the queries asked during this computation. For each $1 \le i \le n$ let $\mathbf{q}'_{\mathsf{dec},i} \leftarrow \mathsf{CHB}_{T_{\mathit{Enc}}}(\mathbf{q}_{\mathsf{dec},i})$. For intuition: this transformation is to express each group element t_j from the ciphertext as a combination of g_j's (which are elements of the public key).

[5]A prior query that should have the same answer as current one.

Add the entries $(\mathbf{q}'_{\mathsf{dec},i}, \mathcal{O}(\mathbf{q}'_{\mathsf{dec},i}))$ to T. Note that each $\mathbf{q}'_{\mathsf{dec},i}$ has non-zero coefficients only for the group elements $(\mathbf{1}, g_1, \ldots, g_n, z_1, \ldots, z_n)$.

Output $\mathrm{SK}'_{\mathrm{ID}} = (w_{\mathrm{ID}}, T, \bar{z}) \in \{0,1\}^*$ where $\bar{z}$ is a concatenation of the elements $z_1, \ldots, z_m$ *as strings.*

***Enc*$'$(PP, ID, b):** Run $C \leftarrow_{\mathrm{R}} \mathit{Enc}(\mathrm{PP}, \mathrm{ID}, b)$ and output C.

***Dec*$'$($\mathbf{SK}'_{\mathbf{ID}}$, C):** Let $\mathrm{SK}'_{\mathrm{ID}}$ be $(w_{\mathrm{ID}}, T, \bar{z})$. Create an empty table T' with the same structure as T (T' will be modified during the computation of *Dec*$'$). Set $\mathrm{SK}_{\mathrm{ID}} = (w_{\mathrm{ID}}, \bar{z})$ and simulate $\mathit{Dec}(\mathrm{SK}_{\mathrm{ID}}, C)$ as follows. For each oracle query $\mathbf{q} = \mathbf{q}^{(\mathbf{1})} + \mathbf{q}^{(\mathsf{z})} + \mathbf{q}^{(\mathsf{t})}$ asked by *Dec* informally we proceed as follows: either (i) determine that an equivalent query was asked before, and use the previous answer, or (ii) use T to represent $\mathbf{q}^{(\mathsf{z})}$ as a sum of g_i's, and then use $\mathcal{O}$ to evaluate the modified query. Formally, first let $R = \mathsf{Eq}(T \cup T')$ be the set of all equality relations among z_i's and g_i's deduced from $T \cup T'$. Note that R contains linear combinations of the form $\mathbf{e} = \mathbf{q}^{\mathbf{1}} + \mathbf{q}^{(\mathsf{z})} + \mathbf{q}^{(\mathsf{g})}$ such that $\mathcal{O}(\mathbf{e}) = 0$. To answer a query $\mathbf{q} = \mathbf{q}^{\mathbf{1}} + \mathbf{q}^{(\mathsf{z})} + \mathbf{q}^{(\mathsf{t})}$ where

$$\mathbf{q}^{(\mathsf{z})} = \sum_{i=1}^{m} b_i z_i \text{ and } \mathbf{q}^{(\mathsf{t})} = \sum_{i=1}^{m} c_i t_i$$

proceed as follows:

Step 1: If $\mathbf{q}^{(\mathsf{z})} \in \mathrm{span}(R^{(\mathsf{z})})$, let $\{c_{\mathbf{e}} \mid \mathbf{e} \in R\}$ be coefficients such that $\mathbf{q}^{(\mathsf{z})} = \sum_{\mathbf{e} \in R} c_{\mathbf{e}} \mathbf{e}^{(\mathsf{z})}$. Set $\hat{\mathbf{q}} = \mathbf{q} - \sum_{\mathbf{e} \in R} c_{\mathbf{e}} \mathbf{e}$. Note that the subtracted value is equal to zero. The modified query $\hat{\mathbf{q}}$ is of the form $\hat{\mathbf{q}} = \hat{\mathbf{q}}^{(\mathbf{1})} + \hat{\mathbf{q}}^{(\mathsf{g})} + \mathbf{q}^{(\mathsf{t})}$ (the coefficients of the group elements z_i are all zero). Therefore, to answer $\mathbf{q}$, query $\alpha \leftarrow \mathcal{O}(\hat{\mathbf{q}})$ and answer $\mathbf{q}$ with α.

Step 2: If there is no way to express $\mathbf{q}^{(\mathsf{z})}$ as a combination of g_i's then check whether a query equivalent to $\mathbf{q}$ was asked before, where equivalence is defined as follows: for every query $\mathbf{q}_{old} \in T' \cup T$ check if $\mathbf{q}^{(\mathsf{z})} \in \mathbf{q}^{(\mathsf{z})}_{old} + \mathrm{span}(R^{(\mathsf{z})})$. If so, let $\{c_{\mathbf{e}} \; ; \; \mathbf{e} \in R\}$ be coefficients such that $\mathbf{q}^{(\mathsf{z})} = \mathbf{q}^{(\mathsf{z})}_{old} + \sum_{\mathbf{e} \in R} c_{\mathbf{e}} \mathbf{e}^{(\mathsf{z})}$. Check if

$$\mathcal{O}(\mathbf{q}^{(\mathbf{1})} + \mathbf{q}^{(\mathsf{t})}) \stackrel{?}{=} \mathcal{O}\left(\mathbf{q}^{(\mathbf{1})}_{old} + \mathbf{q}^{(\mathsf{g})}_{old} + \mathbf{q}^{(\mathsf{t})}_{old} + \sum_{\mathbf{e} \in R} c_{\mathbf{e}}(\mathbf{e}^{(\mathbf{1})} + \mathbf{e}^{(\mathsf{g})})\right) \qquad (1)$$

and if both checks succeeded, answer $\mathbf{q}$ with g where $\{\mathbf{q}_{old} \mapsto g\} \in T' \cup T$.

Step 3: Finally, if the above procedure fails, choose $s \in_{\mathrm{R}} S$, add the entry $\{\mathbf{q} \mapsto s\}$ to T', and return s.

Output: when the simulation of *Dec* concludes, output whatever *Dec* outputs.

It is not hard to see why the above transformation preserves the security of the IBE system. In what follows we bound the loss in the correctness by formalizing the intuition preceding this transformation, i.e. show why the performed tests in the above transformation are necessary and sufficient.

3.2 Analysis of the reduction of the security of $\mathcal{IBE}$ to $\mathcal{RIBE}$

We prove theorem 1. Security is straightforward, and follows from the fact that *KeyGen′* uses *KeyGen* as a black box. Consequently, given an adversary *Adv* that breaks WSS security of $\mathcal{RIBE}$, we can obtain an adversary *Adv′* for $\mathcal{IBE}$ by simulating *Adv*. To answer a private key query made by *Adv* for ID, *Adv′* first queries *KeyGen* to obtain $\mathrm{SK}_{\mathrm{ID}}$, and then simulates *KeyGen′*. We omit the details, and focus on the correctness in the remainder of the section. We prove the following theorem:

Remark: Throughout the analysis we assume that p, the order of the group, is larger than any polynomial in n. We treat the case where p is a polynomial separately in the obvious way. That is, query all the elements of the group. This way, we can compute the discrete logarithms of all elements. Given that we know the whole oracle we can exhaustively check off-line (i.e. after we learn the entire oracle) the possible bits and random strings for *Enc*.

Notational remark: The z_i's group elements depend on the ID. Throughout, this proof we denote by z_i the z_i's which are relevant to the challenge identity which we fix.

The crux of the proof is in showing that *Dec′* is mostly successful in answering oracle queries asked by *Dec*. The argument proceeds in two parts. First, we show that whenever a query is answered according to steps 1 or 2, the answer is determined by previous queries, and that *Dec′* provides the correct answer. Second, we show that when a query is answered according to step 3, then with high probability the answer is undetermined by previous queries, and is in fact close to being uniformly distributed in S. To formalize this we shall consider the following distribution defined for each ID:

$$D_{\mathrm{ID}} \overset{\mathrm{def}}{=} \left\{ \begin{array}{ll} \mathrm{MSK} \leftarrow_{\mathrm{R}} \{0,1\}^n;\ b_0 \leftarrow_{\mathrm{R}} \{0,1\}; & (\mathrm{PP}, T_{\mathsf{setup}}) \leftarrow \mathit{Setup}(\mathrm{MSK}); \\ (\mathrm{SK}_{\mathrm{ID}}, T_{\mathsf{kg}}) \leftarrow \mathit{KeyGen}(\mathrm{MSK}, \mathrm{ID}); & (\mathrm{SK}'_{\mathrm{ID}}, T'_{\mathsf{kg}}) \leftarrow_{\mathrm{R}} \mathit{KeyGen}'(\mathrm{SK}_{\mathrm{ID}}, \mathrm{ID}); \\ (C, T_{\mathsf{enc}}) \leftarrow_{\mathrm{R}} \mathit{Enc}(\mathrm{PP}, \mathrm{ID}, b_0); & (b, T_{\mathsf{dec}}) \leftarrow_{\mathrm{R}} \mathit{Dec}(\mathrm{SK}_{\mathrm{ID}}, C); \end{array} \right\}$$

Recall that the correctness guarantee of $\mathcal{IBE}$ is that $\Pr_{D_{\mathrm{ID}}}[b = b_0] \geq \varepsilon$ for all ID. We also define a distribution D' to be the same as D with the following exception:

$$(b, T_{\mathsf{dec}}) \leftarrow_{\mathrm{R}} \mathit{Dec}(\mathrm{SK}_{\mathrm{ID}}, C) \quad \text{is replaced by} \quad (b, T'_{\mathsf{dec}}) \leftarrow_{\mathrm{R}} \mathit{Dec}'(\mathrm{SK}'_{\mathrm{ID}}, C)$$

where query-answer map T'_{dec} contains the queries asked by the *simulated Dec*, and the simulated answers given by *Dec′* to these queries. Note that the sample spaces of D and D' are identical (as sets). In our analysis we show that D and D' are similar by arguing that the two distributions are close if we restrict our attention to the first i queries in T_{dec} and T'_{dec}.

To further facilitate the comparison of D and D' we define an intermediate distribution D_0 that is the same as D with the following exception: the encoding function σ is generated "on the fly" according to the following rules:

1. All the queries asked during the computation of *Setup*, *KeyGen*, *KeyGen′*, and *Enc* are answered as before. Namely, to answer query $\mathbf{q}$, let T be the set[6] of all query-answer pairs that appeared before $\mathbf{q}$. If $\mathrm{dlog}_{\sigma_T}(\mathbf{q})$ is defined, $\mathbf{q}$ is answered by $\sigma_T(\mathrm{dlog}_{\sigma_T}(\mathbf{q}))$. Otherwise, an element g is chosen randomly from $S \setminus \mathrm{Im}(\sigma_T)$, and $\mathbf{q}$ is answered with g.

[6]Note that this T is not the same as the T in the description of $\mathcal{RIBE}$.

2. For each query $\mathbf{q}$ asked by *Dec* during its simulation by *Dec*$'$, if $\mathrm{dlog}_{\sigma_T}(\mathbf{q})$ is not defined, and for all $\mathbf{q}_{old} \in T$, $\mathbf{q}^{(\mathsf{z})} \notin \mathbf{q}_{old}^{(\mathsf{z})} + \mathrm{span}(R^{(\mathsf{z})})$, an element g is chosen randomly from S, and $\mathbf{q}$ is answered with g. Note that here σ_T may become not 1-1 (thus, not a valid encoding function).

The conceptual difference between D and D_0 is that in D_0 two queries with distinct discrete logarithms may receive the same answer if that answer happens to be chosen randomly twice from S. Notice also that this type of collision can occur in D' when queries are answered according to step 3 of *Dec*$'$. We shall argue that, barring such collisions, D and D_0 are identical with significant probability.

Claim 1. *Let $R = \mathsf{Eq}(T'_{\mathsf{kg}})$, and let $R_{\mathsf{dec}} = \mathsf{Eq}(T'_{\mathsf{kg}} \cup T_{\mathsf{enc}} \cup \mathsf{CHB}_{T_{\mathsf{enc}}}(T_{\mathsf{dec}}))$. Then,*

$$\Pr[R_{\mathsf{dec}} \nsubseteq \mathrm{span}(R)] \leq \frac{1}{n^c}$$

Proof. The claim is shown by a dimension argument. Let V be the vector space of all sums of the form $\mathbf{q}^{(\mathbf{1})} + \mathbf{q}^{(\mathsf{z})} + \mathbf{q}^{(\mathsf{g})}$. There are exactly n coordinates in $\mathbf{q}^{(\mathsf{z})}$ and $\mathbf{q}^{(\mathsf{g})}$, so $\dim(V) = 2n + 1$. Let T_i be the set of queries and answers recorded in T during iteration i in *KeyGen*$'$. Recall that in each iteration a random bit is encrypted and decrypted, and all the queries that are asked in the process are recorded in T. We now observe that the process of encrypting and decrypting the bit b in D is identical to a single round of the repetitive sampling in *KeyGen*$'$, and so the set $T_{\mathsf{enc}} \cup T_{\mathsf{dec}}$ has the same distribution as each of the sets T_i. It is therefore sufficient to find the probability of the event $\mathsf{new}_k = \{R_{k+1} \nsubseteq \mathrm{span}(R_{n^c} \cup \cdots \cup R_{2n^c})\}$, where $k \in_{\mathrm{R}} \{n^c, \ldots, 2n^{2c}\}$.

Note that for all i, $R_1 \cup \cdots \cup R_i \subseteq V$ and that new_k implies $\dim(R_1 \cup \cdots \cup R_{k+1}) > \dim(R_1 \cup \cdots \cup R_k)$. Since $\dim(V) = 2n+1$, there may be at most that many indices i for which new_i occurs. Consequently, by choosing k randomly, we hit such an i with probability at most $\frac{2n+1}{n^{2c}}$. That is, $\Pr[\mathsf{new}_k] \leq \frac{1}{n^c}$. □

Claim 2. *Fix an encoding function σ, and values MSK, PP, SK$_{ID}$, and let S_δ be the set of $x \in \mathbb{Z}_p$ such that*

$$\Pr\left[x \in \mathrm{dlog}(T_{\mathsf{enc}} \cup T_{\mathsf{dec}})\ ;\ \begin{array}{c}(C, T_{\mathsf{enc}}) \leftarrow_R Enc(PP, ID, b_0)\\ (b, T_{\mathsf{dec}}) \leftarrow_R Dec(SK_{ID}, C)\end{array}\right] \geq \delta$$

Then, $\Pr[S_{1/n^{c-1}} \subseteq \mathrm{dlog}(T)] \geq 1 - n/e^n$ *where the random variable T is the table constructed by KeyGen$'$(SK$_{ID}$, ID).*

Proof. We start with a counting argument to show that there are at most n/δ elements in S_δ. Let Y be the set of strings $\{(b, r_{\mathsf{enc}}, r_{\mathsf{dec}})\}$ where b is the bit to be encrypted, and r_{enc} and r_{dec} are the randomness used by *Enc* and *Dec* respectively. For each $y \in T$ let $\mathrm{dlog}(y) := \mathrm{dlog}(T_{\mathsf{enc}} \cup T_{\mathsf{dec}})$ (note that $\mathrm{dlog}(y)$ is a set notation) where T_{enc} and T_{dec} are the query-answer sets generated by $Enc(\mathrm{PP}, \mathrm{ID}, b; r_{\mathsf{enc}})$ and $Dec(\mathrm{SK}_{\mathrm{ID}}, C; r_{\mathsf{dec}})$. Then, on the one hand we have $|Y| \geq |\{y \in Y | \mathrm{dlog}(y) \cap S_\delta \neq \emptyset\}|$. On the other hand, for each $x \in S_\delta$, there are at least $\delta|Y|$ values y such that $x \in \mathrm{dlog}(y)$, and for each y it holds that $|\mathrm{dlog}(y)| \leq n$. Consequently, we obtain

$$|Y| \geq |\{y \in Y | \mathrm{dlog}(y) \cap S_\delta \neq \emptyset\}| \geq \delta|Y| \cdot |S_\delta|/n$$

and so $n/\delta \geq |S_\delta|$. Now, consider the first n^c iterations of the loop in *KeyGen*$'$. For each $x \in S_{1/n^{c-1}}$ we have $\Pr[x \notin \mathrm{dlog}(T)] \leq (1 - 1/n^{c-1})^{n^c} \leq 1/e^n$. Applying the union bound for all such x we get $\Pr[S_{1/n^{c-1}} \nsubseteq \mathrm{dlog}(T)] \leq n/e^n$. □

Lemma 1. *Let $\delta \leftarrow_R D$, let $R = \mathsf{Eq}(T'_{\mathsf{kg}})$, and let $\mathbf{q}_1, \ldots, \mathbf{q}_n$ be the queries in T_{dec}. For each $1 \leq i \leq n$ let*

$$\begin{aligned} F_i &= \{\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{setup}} \cup T_{\mathsf{kg}} \cup T_{\mathsf{enc}} \cup T_{\mathsf{dec},i-1})\} \\ E_{1,i} &= \left\{\mathbf{q}_i^{(\mathsf{z})} \in \mathrm{span}(R^{(\mathsf{z})})\right\} \\ E_{2,i} &= \left\{\exists \mathbf{q} \in T'_{i-1} \text{ s.t. } \mathbf{q}_i^{(\mathsf{z})} \in \mathbf{q}^{(\mathsf{z})} + \mathrm{span}\left(R^{(\mathsf{z})}\right) \text{ and } \mathrm{dlog}(\mathbf{q}_i) = \mathrm{dlog}(\mathbf{q})\right\} \end{aligned}$$

and $\mathsf{fail}_i = F_{1,i} \wedge \neg(E_{1,i} \vee E_{2,i})$. Then, $\Pr[\mathsf{fail}_i] \leq \frac{n}{e^n} + \frac{1}{n^c} + \frac{6n}{n^{c-1}}$.

Proof. We consider the following subcases of F_i separately: $F_{1,i} = \{\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{setup}} \cup T_{\mathsf{kg}})\}$, $F_{2,i} = \{\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{enc}})\}$, and $F_{3,i} = \{\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{dec},i-1})\}$. Then $F_i = F_{1,i} \cup F_{2,i} \cup F_{3,i}$.

Case 1: event $F_{1,i}$ occurs. Fix $W = \mathrm{dlog}(T_{\mathsf{setup}} \cup T_{\mathsf{kg}})$, and consider any $y \in W$, and let $S = S_{n^{c-1}}$ be the set defined in Claim 2. Then we can write

$$\begin{aligned} \Pr[\mathrm{dlog}(q_i) \in W \wedge \neg E_{2,i}] &= \sum_{y \in W} \Pr[\mathrm{dlog}(q_i) = y \wedge \neg E_{2,i}] \\ &= \sum_{y \in W \cap S} \Pr[\mathrm{dlog}(q_i) = y \wedge \neg E_{2,i}] + \sum_{y \in W \setminus S} \Pr[\mathrm{dlog}(q_i) = y \wedge \neg E_{2,i}] \\ &\leq \sum_{y \in W \cap S} \Pr[\mathrm{dlog}(q_i) = y \wedge \neg E_{2,i}] + \frac{2n}{n^{c-1}} \qquad (2) \end{aligned}$$

Where the last inequality follows from the definition of S and the fact that $|W| \leq 2n$. Let us now focus on the remaining probability. For the moment we only consider $y \in W$. We can therefore restrict out attention to an event $\neg E'_{2,i}$ where $\Pr[\neg E'_{2,i}] \geq \Pr[\neg E_{2,i}]$, and $\neg E'_{2,i}$ is defined as the union of the following two events:

$$B = \{y \notin \mathrm{dlog}(T)\} \qquad \text{and} \qquad B' = \left\{\exists q \in T \text{ s.t. } \mathrm{dlog}(q) = y \text{ and } \mathbf{q}_i^{(\mathsf{z})} \notin \mathbf{q}^{(\mathsf{z})} + \mathrm{span}(R^{(\mathsf{z})})\right\}$$

Since $y \in S$, we get from Claim 2 that $\Pr[B] \leq n/e^n$. To bound the probability of event B' note that $\mathbf{q}_i^{(\mathsf{z})} \notin \mathbf{q}^{(\mathsf{z})} + \mathrm{span}(R^{(\mathsf{z})})$ implies $\mathbf{q}_i - \mathbf{q} \notin \mathrm{span}(R)$. However, since $\mathrm{dlog}(\mathbf{q}_i) = \mathrm{dlog}(\mathbf{q})$ we have that $\mathbf{q}_i - \mathbf{q} \in R_{\mathsf{dec}}$. Therefore, by Claim 1, $\Pr[B'] \leq \frac{1}{n^c}$. Getting back to our calculation of (2) we get:

$$\begin{aligned} \Pr[\mathrm{dlog}(q_i) \in W \wedge \neg E_{2,i}] &\leq \sum_{y \in W \cap S} \Pr[\mathrm{dlog}(q_i) = y \wedge (B \vee B')] + \frac{2n}{n^{c-1}} \qquad (3) \\ &\leq \frac{n}{e^n} + \frac{1}{n^c} + \frac{2n}{n^{c-1}} \qquad (4) \end{aligned}$$

Case 2: event $F_{2,i}$ occurs. In this case we have $\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{enc}})$. Let $\mathbf{q} \in T_{\mathsf{enc}}$ be a query such that $\mathrm{dlog}(\mathbf{q}) = \mathrm{dlog}(\mathbf{q}_i)$. Then $\mathbf{q}_i - \mathbf{q} \in R_{\mathsf{dec}}$, and by Claim 1, $\Pr[\mathbf{q}_i - \mathbf{q} \notin \mathrm{span}(R)] \leq \frac{2n}{n^{c-1}}$. The following is a crucial point in the argument: recall that for all $\mathbf{q} \in T_{\mathsf{enc}}$, $\mathbf{q}^{(\mathsf{z})} = 0$. Therefore, $\mathbf{q}_i - \mathbf{q} \in \mathrm{span}(R)$ implies $\mathbf{q}_i^{(\mathsf{z})} \in \mathrm{span}(R^{(\mathsf{z})})$. Consequently, we get

$$\Pr[\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{enc}}) \wedge \neg E_{1,i}] \leq \frac{2n}{n^{c-1}} \qquad (5)$$

Case 3: event $F_{3,i}$ occurs. In this case we have $\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{dec},i-1})$. First, suppose that $\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T'_{i-1})$, and let $\mathbf{q} \in T'_{i-1}$ such that $\mathrm{dlog}(\mathbf{q}_i) = \mathrm{dlog}(\mathbf{q})$. This implies that $\mathbf{q}_i - \mathbf{q} \in R_{\mathsf{dec}}$, and therefore $\neg E_{1,i}$ holds only if $\mathbf{q}_i - \mathbf{q} \notin \mathrm{span}(R)$. Second, suppose that $\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T'_{\mathsf{dec},i-1}) \setminus \mathrm{dlog}(T'_{i-1})$. Then, by definition of T'_{i-1}, either $\mathbf{q}_i^{(\mathsf{z})} \in \mathrm{span}(R^{(\mathsf{z})})$ (the part of the query that involves z_i's can be rewritten as a combination of g_i's), or there is a query $\mathbf{q} \in T'_{i-1}$ and coefficients $\{c_{\mathbf{e}} \; ; \; \mathbf{e} \in R\}$ such that $\mathbf{q}^{(\mathsf{z})} = \mathbf{q}_{old}^{(\mathsf{z})} + \sum_{\mathbf{e}\in R} c_{\mathbf{e}}\mathbf{e}^{(\mathsf{z})}$ and equality (1) from *Dec'* holds. But this implies

$$\mathcal{O}(\mathbf{q}_i) = \mathcal{O}(\mathbf{q}_i^{(\mathbf{1})} + \mathbf{q}_i^{(\mathsf{z})} + \mathbf{q}_i^{(\mathsf{t})}) = \mathcal{O}(\mathbf{q}_i^{(\mathbf{1})} + \mathbf{q}_i^{(\mathsf{t})} + \mathbf{q}_{old}^{(\mathsf{z})} + \sum_{\mathbf{e}\in R} c_{\mathbf{e}}\mathbf{e}^{(\mathsf{z})}) \tag{6}$$

$$= \mathcal{O}(\mathbf{q}_{old}^{(\mathbf{1})} + \mathbf{q}_{old}^{(\mathsf{g})} + \mathbf{q}_{old}^{(\mathsf{t})} + \sum_{\mathbf{e}\in R} c_{\mathbf{e}}(\mathbf{e}^{(\mathbf{1})} + \mathbf{e}^{(\mathsf{g})}) + \mathbf{q}_{old}^{(\mathsf{z})} + \sum_{\mathbf{e}\in R} c_{\mathbf{e}}\mathbf{e}^{(\mathsf{z})}) \tag{7}$$

$$= \mathcal{O}(\mathbf{q}_{old}^{(\mathbf{1})} + \mathbf{q}_{old}^{(\mathsf{g})} + \mathbf{q}_{old}^{(\mathsf{t})} + \mathbf{q}_{old}^{(\mathsf{z})}) = \mathcal{O}(\mathbf{q}) \tag{8}$$

Equality (7) holds because condition (1) from *Dec'* holds. The first equality in (8) holds because for all $\mathbf{e} \in \mathrm{span}(R)$, $\mathcal{O}(\mathbf{e}) = 0$ and because $\mathcal{O}$ is linear. We have shown $\mathrm{dlog}(\mathbf{q}_i) = \mathrm{dlog}(\mathbf{q})$, but this contradicts our assumption that $\mathrm{dlog}(\mathbf{q}_i) \notin \mathrm{dlog}(T'_{i-1})$. Thus, we have:

$$\Pr[(\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{dec},i-1})) \wedge \neg(E_{1,i} \vee E_{2,i})] \leq \frac{2n}{n^{c-1}} \tag{9}$$

Combining (4), (5), and (9) we get: $\Pr[\mathsf{fail}_i] \leq \frac{n}{e^n} + \frac{1}{n^c} + \frac{6n}{n^{c-1}}$. □

Lemma 2. *Let E be any event in the sample space* $\mathrm{supp}(D) \cup \mathrm{supp}(D_0)$. *Then,*

$$|\Pr_{\delta \leftarrow_R D}[\delta \in E] - \Pr_{\delta \leftarrow_R D_0}[\delta \in E]| \leq 4n/p$$

Proof. We use a coupling argument to show that D and D_0 are close. Consider the following alternative method for generating distribution D. Run the process for sampling from D_0. If at any point during the computation of *Dec* an assignment $\sigma(\mathrm{dlog}(\mathbf{q}_i)) \leftarrow_{\mathrm{R}} S$ is performed, and

$$\mathrm{dlog}(\mathbf{q}_i) \in \mathrm{dlog}(T_{\mathsf{setup}} \cup T_{\mathsf{kg}} \cup T'_{\mathsf{kg}} \cup T_{\mathsf{enc}} \cup T_{\mathsf{dec},i-1}) \tag{10}$$

holds, then resample $\sigma(\mathrm{dlog}(\mathbf{q}_i)) \leftarrow_{\mathrm{R}} S$ until (10) no longer holds. That is, instead of keeping the first sampled value, we keep sampling random strings in S until no collision occurs. Since the only difference between D_0 and D is that the answer to q_i in D is random among all values that do not cause a collision, it is clear that this alternate procedure generates exactly distribution D.

Now we run the procedure for generating D_0 and the alternate procedure for D in a coupled manner up to the first query $\mathbf{q}_i$ for which (10) holds. Then, conditioned on (10) never occurring, E occurs in D_0 if and only if it occurs in D. Let F be the event that (10) holds at some point during the generation of D. Then we obtain $\Pr_{\delta\leftarrow_{\mathrm{R}} D}[\delta \in E|\neg F] = \Pr_{\delta\leftarrow_{\mathrm{R}} D_0}[\delta \in E|\neg F]$, and

$$\begin{aligned}
\Pr_{\delta\leftarrow_{\mathrm{R}} D}[\delta \in E] &= \Pr_{\delta\leftarrow_{\mathrm{R}} D}[\delta \in E|F]\Pr[F] + \Pr_{\delta\leftarrow_{\mathrm{R}} D}[\delta \in E|\neg F]\Pr[\neg F] \\
&= \Pr_{\delta\leftarrow_{\mathrm{R}} D}[\delta \in E|F]\Pr[F] + \Pr_{\delta\leftarrow_{\mathrm{R}} D_0}[\delta \in E] - \Pr_{\delta\leftarrow_{\mathrm{R}} D_0}[\delta \in E|F]\Pr[F] \\
&= \Pr[F]\left(\Pr_{\delta\leftarrow_{\mathrm{R}} D}[\delta \in E|F] - \Pr_{\delta\leftarrow_{\mathrm{R}} D_0}[\delta \in E|F]\right) + \Pr_{\delta\leftarrow_{\mathrm{R}} D_0}[\delta \in E]
\end{aligned}$$

which immediately implies the conclusion. □

Lemma 3. *Let E be any event in the sample space* $\operatorname{supp}(D_0) \cup \operatorname{supp}(D')$*. Then,*

$$|\Pr_{\delta \leftarrow_R D_0}[\delta \in E] - \Pr_{\delta \leftarrow_R D'}[\delta \in E]| \leq 1/n^{c/2}$$

Proof. Again, we apply a coupling argument to show that D_0 and D' are close. Consider the following alternative way of generating distribution D_0. Proceed as D' up to the first query i of the *simulated Dec* for which the event $\mathsf{fail}_i = F_i \wedge \neg(E_{1,i} \vee E_{2,i})$ occurs. If fail_i has occurred then answer query q_j for $j \geq i$ (that is, including query q_i) as follows: if F_j occurs then reply $\sigma(\operatorname{dlog}(q_j))$.

To show that this procedure generates D_0 we must argue that (i) unless fail_i occurs for some i, the algorithm *Dec*$'$ always replies with $\sigma(\operatorname{dlog}(q_j))$ for all $1 \leq j \leq n$ where F_j occurs, and (ii) when F_j and $E_{1,j}$ do not hold, *Dec*$'$ replies with a random element of S.

We first show property (i): consider an index j for which F_j occurs, and yet fail_i has not occurred for $1 \leq i \leq j$. This means that either $E_{1,j}$ or $E_{2,j}$ must also occur. We deal with each case separately.

Case 1: $E_{1,j}$ occurs. Then $\mathbf{q}_j^{(\mathsf{z})} \in \operatorname{span}(R^{(\mathsf{z})})$ so there exist $d_1, \ldots, d_l$ such that $\mathbf{q}_j^{(\mathsf{z})} = \sum_{i=1}^{l} d_i \mathbf{e}_i^{(\mathsf{z})}$. Consider a new query $\hat{\mathbf{q}} = \mathbf{q}_j - \sum_{i=1}^{l} d_i \mathbf{e}_i$. Then we have

$$\operatorname{dlog}(\hat{\mathbf{q}}) = \operatorname{dlog}(\mathbf{q}_j) - \operatorname{dlog}(d_1\mathbf{e}_1 + \cdots + d_l\mathbf{e}_l) = \operatorname{dlog}(\mathbf{q}_j) \quad \text{and}$$
$$\hat{\mathbf{q}}^{(\mathsf{z})} = \mathbf{q}_j^{(\mathsf{z})} - (d_1\mathbf{e}_1^{(\mathsf{z})} + \cdots + d_l\mathbf{e}_l^{(\mathsf{z})}) = 0$$

In this case *Dec*$'$ follows step 1 by querying $\alpha \leftarrow \mathcal{O}(\hat{\mathbf{q}})$, and returning $\alpha = \sigma(\operatorname{dlog}(\hat{\mathbf{q}}))$ as the answer to $\mathbf{q}_j$.

Case 2: $E_{2,j}$ occurs. Then, there is a query $\mathbf{q} \in T'_{j-1}$ such that $\mathbf{q}_i^{(\mathsf{z})} \in \mathbf{q} + \operatorname{span}(R^{(\mathsf{z})})$ and $\operatorname{dlog}(\mathbf{q}_j) = \operatorname{dlog}(\mathbf{q})$. Then, there exist $d_1, \ldots, d_l$ such that $\mathbf{q}_j^{(\mathsf{z})} = \mathbf{q}^{(\mathsf{z})} + \sum_{i=1}^{l} d_i \mathbf{e}_i^{(\mathsf{z})}$. We can then write

$$\mathcal{O}(\mathbf{q}_j) = \mathcal{O}\left(\mathbf{q}_j^{(\mathsf{1})} + \mathbf{q}^{(\mathsf{z})} + \sum_{i=1}^{l} d_i \mathbf{e}_i^{(\mathsf{z})} + \mathbf{q}_j^{(\mathsf{t})}\right) \text{ and } \mathcal{O}(\mathbf{q}) = \mathcal{O}\left(\mathbf{q}^{(\mathsf{1})} + \mathbf{q}^{(\mathsf{z})} + \sum_{i=1}^{l} d_i \mathbf{e}_i^{(\mathsf{z})} + \mathbf{q}^{(\mathsf{g})} + \mathbf{q}^{(\mathsf{t})}\right) \tag{11}$$

Using the fact that $\operatorname{dlog}(\mathbf{q}_j) = \operatorname{dlog}(\mathbf{q})$ we get $\mathcal{O}(\mathbf{q}_j) = \mathcal{O}(\mathbf{q})$ which together with (11) implies $\mathcal{O}(\mathbf{q}_j^{(\mathsf{1})} + \mathbf{q}_j^{(\mathsf{t})}) = \mathcal{O}(\mathbf{q}^{(\mathsf{1})} + \mathbf{q}^{(\mathsf{g})} + \mathbf{q}^{(\mathsf{t})})$. Therefore, all the conditions of step 2 in *Dec*$'$ hold and the query is answered with g where $\{\mathbf{q} \mapsto g\} \in T'_{j-1}$. Assuming that all queries prior to $\mathbf{q}_j$ were answered correctly we get $g = \sigma(\operatorname{dlog}(\mathbf{q})) = \sigma(\operatorname{dlog}(\mathbf{q}_j))$.

To show property (*ii*) we must argue that when F_j and $E_{1,j}$ do not hold *Dec*$'$ will not enter step 2, and will proceed to answer the query according to step 3. Clearly if $E_{1,j}$ does not hold the algorithm will not enter step 1. Suppose that *Dec*$'$ answers $\mathbf{q}_j$ according to step 2. Then there exists a query $\mathbf{q} \in T'_{j-1}$, and linear combination of $\mathbf{e}_i$'s such that $\mathbf{q}_j^{(\mathsf{z})} = \mathbf{q}^{(\mathsf{z})} + \sum_{i=1}^{l} d_i \mathbf{e}_i^{(\mathsf{z})}$, and condition (1) holds. However, in this case (11) implies that F_j must hold, a contradiction.

To conclude the proof consider running the procedure for generating D' and the alternative procedure for D_0 in a coupled manner until the first i for which fail_i occurs. At that point each procedure continues on its own. Clearly, if fail_i does not occur for any i then event E occurs in D_0 if and only if it occurs in D'. From Lemma 1 and Lemma 2 we know that for each i, $\Pr_{D'}[F_{1,i} \wedge \neg(E_{1,i} \vee E_{2,i})] \leq \frac{n}{e^n} + \frac{1}{n^c} + \frac{6n}{n^{c-1}} + 4n/p$. Applying the union bound we obtain $\Pr_{D'}[\mathsf{fail}_1 \vee \cdots \vee \mathsf{fail}_n] \leq \frac{n}{e^n} + \frac{1}{n^c} + \frac{6n}{n^{c-1}} + 4n/p \leq \frac{1}{n^{c-3}} \leq \frac{1}{n^{c/2}}$. The conclusion follows. □

4 The Attack on Restricted IBE

We describe an attack on any Identity Based Encryption scheme where individual private keys do not contain any group elements. Our adversary recovers the plaintext of a challenge ciphertext with a probability that is close to the correctness guarantee of the restricted IBE.

Theorem 2. *Let $\mathcal{RIBE} = \langle Setup, KeyGen, Enc, Dec \rangle$ be a restricted IBE with parameters p, m, n, ε. Then, for every $c > 0$ and sufficiently large n, there exists an adversary which breaks the security of the $\mathcal{RIBE}$ with $\mathsf{poly}(m, n)$ oracle queries and advantage $\varepsilon - \frac{1}{2} - \frac{1}{n^c}$.*

Our adversary is determined by the parameters c_1, c_2, c_3, c_4, which when set to $c_1 = c_2 = c_4 = c + 3$ and $c_3 = 2c + 4$ yield the theorem.

Sampling procedures We use the following two sampling procedures. The first procedure $\mathsf{SampleED}(\mathrm{PP}, \mathrm{ID}, \mathrm{SK}_{\mathrm{ID}})$, where ED stands for Enc-Dec, samples query transcripts by encrypting and decrypting many random bits: for $1 \leq i \leq n^{c_4}$ choose $b_i \in_{\mathrm{R}} \{0, 1\}$; $r_i \leftarrow_{\mathrm{R}} \mathsf{Rand}(Enc)$, where $\mathsf{Rand}(Enc)$ denotes a uniformly random string of the appropriate length for Enc; compute $C_i \leftarrow_{\mathrm{R}} Enc(\mathrm{PP}, \mathrm{ID}, b_i; r_i)$; and compute $Dec(\mathrm{SK}_{\mathrm{ID}}, C_i)$. Let T_{enc} be the set of query-answer pairs that appear during all the encryptions. For each query $\mathbf{q} = \mathbf{q^{(1)}} + \mathbf{q^{(g)}} + \mathbf{q^{(t)}}$ of the decryption algorithm, let $\hat{\mathbf{q}} = \mathsf{CHB}_{T_{enc}}(\mathbf{q^{(t)}})$. Add the entry $\{\hat{\mathbf{q}} \mapsto \mathcal{O}(\mathbf{q})\}$ to a set T_{dec}. The output of $\mathsf{SampleED}$ is $((b_1, r_1), \ldots, (b_{n^{c_4}}, r_{n^{c_4}}), T_{enc}, T_{dec})$. Note that we do not include the ciphertexts in the output because they are determined by b_i, r_i, and the query-answer pairs recorded in T_{enc} and T_{dec}. In the second sampling procedure $\mathsf{SampleEnc}(\mathrm{PP}, \mathrm{ID})$ we only sample the encryption: choose $b \in_{\mathrm{R}} \{0, 1\}$; $r \leftarrow_{\mathrm{R}} \mathsf{Rand}(Enc)$; and compute $C \leftarrow_{\mathrm{R}} Enc(\mathrm{PP}, \mathrm{ID}, b; r)$. Let T_{enc} be the set of queries and answers that appear during the encryption. The output of $\mathsf{SampleEnc}$ is (b, r, T_{enc}).

Initialize. Randomly choose $k_1, k_2 \in_{\mathrm{R}} \left\{n^{2c}, \ldots, 2n^{2c}\right\}$, let $l = k_1 + k_2$, $\mathrm{ID}_*, \mathrm{ID}_1, \ldots, \mathrm{ID}_l \in_{\mathrm{R}} \mathcal{ID}$. Submit ID_* as the challenge identity, and $\mathrm{ID}_1, \ldots, \mathrm{ID}_{k_1}$ as the identities for which the adversary wishes to obtain private keys. Note that the adversary requests private keys only for the first k_1 identities among the l that were chosen. The adversary is then given public parameters $\mathrm{PP} = (w_{\mathrm{PP}}, g_1, \ldots, g_m)$, private keys $\mathrm{SK}_i \in \{0, 1\}^n$ for ID_i, $1 \leq i \leq l$, and a challenge ciphertext C_*.

Step 1: Learning linear relations among public key group elements.
For each $1 \leq i \leq k_1$, run $\mathsf{SampleED}(\mathrm{PP}, \mathrm{ID}_i, \mathrm{SK}_i)$
to obtain a tuple
$\gamma^{(i)} = ((b_1^{(i)}, r_1^{(i)}), \ldots, (b_{n^{c_4}}^{(i)}, r_{n^{c_4}}^{(i)}), T_{enc}^{(i)}, T_{dec}^{(i)})$.

Step 2: Learning frequently accessed elements of $\mathbb{Z}_p$. For each $k_1 + 1 \leq i \leq l$, run $\mathsf{SampleEnc}(\mathrm{PP}, \mathrm{ID}_i)$ to obtain a tuple $\delta^{(i)} = (b^{(i)}, r^{(i)}, T_{enc}^{(i)})$.

Step 3: Generating a fake private key for ID_*. Choose $k_3 \in_{\mathrm{R}} \left\{n^{2c}, \ldots, 2n^{2c}\right\}$, and repeat the following online and offline phases k_3 times. For consistency with previous steps we index each repetition with $l + 1 \leq j \leq l + k_3$:

Offline phase j (note that the numbering of the phases starts from $l + 1$). Let *view* be the view of the adversary at this point. The view includes the public key PP, the identities $\mathrm{ID}_1, \ldots, \mathrm{ID}_l, \mathrm{ID}_*$, the k_1 many outputs of $\mathsf{SampleED}$ and the k_2 many outputs of $\mathsf{SampleEnc}$,

and the sets
$(T_{enc}^{(l+1)}, T_{dec}^{(l+1)}), \dots, (T_{enc}^{(j-1)}, T_{dec}^{(j-1)})$ that are defined in the phase below.
We define a distribution D on master secret keys and transcripts of *Setup* and *KeyGen* as follows. For every $(\alpha, \beta) \in \{0,1\}^n \times (\mathbb{Z}_p[G] \times S)^{2m}$ we set

$$\Pr[(\alpha,\beta)] \overset{\text{def}}{=} \Pr\left[\begin{array}{cc|c} \text{MSK} = \alpha, & \text{MSK} \in_{\text{R}} \{0,1\}^n, & \\ \underbrace{T_{setup} \cup T_{kg}}_{T} = \beta; & (\text{PP}, T_{setup}) \leftarrow \mathit{Setup}(\text{MSK}), & \mathit{view} \\ & (\text{SK}_*, T_{kg}) \leftarrow \mathit{KeyGen}(\text{MSK}, \text{ID}_*) & \end{array}\right]$$

and sample $(\text{MSK}, T) \leftarrow_{\text{R}} D$. Note that here PP, MSK, SK_* denote the sampled ("fake") world. Let σ_T be the partial encoding function defined by T; and note that this σ_T is well-defined. We now generate a private key for ID_* by evaluating $\mathit{KeyGen}(\text{MSK}, \text{ID}_*)$. Note that all the queries asked by *KeyGen* can be answered using T_{kg}. Let SK_* be the output of the simulated *KeyGen*.

Online phase j. Initialize an empty set $T_{dec}^{(j)}$ of query answer pairs. Choose $b_1, \dots, b_{n^{c_4}} \in_{\text{R}} \{0,1\}$, and compute $(C_i, T_{enc,i}) \leftarrow_{\text{R}} \mathit{Enc}(\text{PP}, \text{ID}_*, b_i)$. Let
$T_{enc}^{(j)} = \bigcup_{i=1}^{n^{c_4}} T_{enc,i}$ be the set of all query-answer pairs that appear during the generation of $C_1, \dots, C_{n^{c_4}}$. Then, simulate $\mathit{Dec}(\text{SK}_*, C_i)$ for $1 \leq i \leq n^{c_4}$ as follows. For each oracle query $\mathbf{q}$ during decryption of C_i, let $\hat{\mathbf{q}} = \mathsf{CHB}_{T_{enc,i}}(\mathbf{q})$. The modified query is of the form $\hat{\mathbf{q}} = \hat{\mathbf{q}}^{(\mathbf{1})} + \hat{\mathbf{q}}^{(\mathbf{g})}$. If $\sigma_T(\text{dlog}(\hat{\mathbf{q}}))$ is defined, respond to $\mathbf{q}$ with $\sigma_T(\text{dlog}(\hat{\mathbf{q}}))$. Otherwise, query the actual oracle, and respond with $\mathcal{O}(\mathbf{q})$. In both cases, add the pair $(\hat{\mathbf{q}}, \mathcal{O}(\mathbf{q}))$ to $T_{dec}^{(j)}$.

Note that all the oracle queries of *KeyGen* are of the form $\mathbf{q} = a\mathbf{1}$ for some $a \in \mathbb{Z}_p$. Therefore, the adversary can determine whether $\sigma_T(\mathbf{q})$ is defined by checking if $\sigma_T(a)$ is defined. Clearly, this will no longer be true in the next phase, when we attempt to decrypt the challenge ciphertext.

Step 4: Preparing to decrypt the challenge. Perform another iteration of the offline and online phases of step 3 with the exception that in the online phase, only $n^{c_4} - 1$ ciphertexts are encrypted and decrypted. Let T_{dec}^{step4} be the set of query-answer pairs constructed during decryption. We think of the last encryption of the online phase to be the actual encryption that generated the challenge ciphertext C_*.

Step 5: The decryption procedure. Recall that in $\mathcal{RIBE}$ the SK's do not contain any group elements. To decrypt the challenge ciphertext the adversary simulates $\mathit{Dec}(\text{SK}_*, C_*)$ (**where SK_* is the private key generated in step 4**), and answers oracle queries as follows: let $\mathbf{q} = \mathbf{q}^{(\mathbf{1})} + \mathbf{q}^{(\mathbf{g})} + \mathbf{q}^{(\mathbf{t})}$ be a query. The adversary queries the oracle to obtain $\alpha = \mathcal{O}(\mathbf{q})$. If there exists a query $\mathbf{q}'$ such that $\{\mathbf{q}' \mapsto \alpha\} \in T_{dec}^{\mathsf{step4}}$, then the adversary responds to $\mathbf{q}$ with $\sigma_T(\mathbf{q}')$, if it is defined, and with α otherwise. Finally, the adversary outputs what *Dec* outputs.

5 Overview of the analysis of the attack

We will show that our adversary succeeds in decrypting the challenge ciphertext with probability arbitrarily close to the correctness probability of the IBE. The full details of this argument are complex. We start by presenting a high level, complete, description of our analysis, whereas the actual proof is given in Section 6.

The strategy of our adversary can be summarized as follows: the adversary first performs various operations to learn information about the encoding function σ, to which he has access through the oracle $\mathcal{O}$. The adversary then constructs, in an offline manner (without accessing the oracle), a partial encoding function σ_T which is consistent with what the adversary has learned about σ, and a master secret key MSK. Note that σ_T is defined only on a small subset of $\mathbb{Z}_p$ (just enough to compute *Setup*(MSK) without making any queries to the actual oracle), and most likely disagrees with the actual encoding function on some elements in that subset. Similarly, the master secret key MSK generated by the adversary is only consistent with adversary's view, but may very well be different from the actual master secret key that was used by the challenger.

After choosing σ_T and MSK our adversary should be viewed as trying to generate a private key for ID_*, and then decrypting the challenge ciphertext, using an encoding function $\hat{\sigma}$ which overlays σ_T on top of σ. Although we will not refer to $\hat{\sigma}$ directly in the formal part of our analysis, we shall use it in the current (informal) part to illustrate the difficulties and pitfalls that we must avoid. More specifically, the encoding function that the adversary attempts to present to *KeyGen* and *Dec* is defined as $\hat{\sigma}(x) = \sigma_T(x)$ for all $x \in \mathbb{Z}_p$ on which σ_T is defined, and $\hat{\sigma}(x) = \sigma(x)$ for all the remaining inputs. Before proceeding, we note that $\hat{\sigma}$ may contain collisions (i.e. $x \neq y$ such that $\hat{\sigma}(x) = \hat{\sigma}(y)$), and is therefore not a valid encoding function. However, such collisions are extremely unlikely to be discovered, and we shall ignore them in this informal description of our analysis.

To show that our adversary succeeds with high probability it suffices to show three things: (i) the adversary successfully simulates *KeyGen* and *Dec* using $\hat{\sigma}$ (we will explain why this is non-trivial in a moment), (ii) the challenge ciphertext is generated by the challenger using σ. Thus, we must show that for all queries asked by the encryption algorithm during the generation of the challenge ciphertext, $\hat{\sigma}$ and σ are identical. Finally, we must show that (iii) the distribution of the view of the decryption algorithm during its simulation by the adversary is close to the view it would have if all values were generated honestly by a party that knows the entire correct encoding function and the actual master secret key. The last property requires the first two to hold (but needs additional treatment), and also relies on the correctness property of the IBE. We next informally describe how our adversary achieves each of the above three properties.

Exposing an alternate encoding function $\hat{\sigma}$. Let us first consider the computation of *KeyGen*(MSK, ID_*) during step 4. The only group element given to *KeyGen* as input is the identity $\mathbf{1}$. Therefore, for all oracle queries $\mathbf{q} = a\mathbf{1}$ asked by *KeyGen*, we have $\mathrm{dlog}_{\sigma_T}(\mathbf{q}) = \mathrm{dlog}_{\sigma}(\mathbf{q})$. Therefore, all we need is to observe that our adversary responds with $\sigma_T(a)$ when it is defined, and with $\sigma(a)$ otherwise.

The case of decryption is much more complex. One difficulty stems from the fact that now oracle queries are of the form $\mathbf{q} = \mathbf{q^{(1)}} + \mathbf{q^{(g)}} + \mathbf{q^{(t)}}$, where t_i are the group elements in the ciphertext. The problem is that even though each t_i was generated by *Enc* as a linear combination of the group elements in the public key (the g_i's), the adversary does not know these linear combinations since the encryption was computed by the challenger, and so is unable to compute $\mathrm{dlog}_{\sigma_T}(\mathbf{q})$ (recall that $\mathrm{dlog}_{\sigma_T}(\cdot)$ is defined only for $g \in S \cap \mathrm{Im}(\sigma_T)$).

To address this problem, in Step 5, we first query the actual oracle on $\mathbf{q}$ to obtain a value α, and then check if we have already seen α as an answer to an oracle query in the past. Since for all past encryption-decryption computations we know how to represent the t_i's as linear combinations of g_i's, if we find such a combination that maps to α we are able to check whether $\sigma_T(\mathbf{q})$ is defined, and give the correct answer. If we haven't seen α as an answer to a query then, unless a new linearly independent equality could be discovered during the encryption-decryption of the challenge, $\sigma_T(\mathbf{q})$

is undefined, and it is safe to respond with the newly obtained random value α.

We note that showing that it is unlikely that a new linearly independent relation will appear in Step 5 is quite subtle because the way oracle queries of *Dec* are answered is different from Step 3, which may in turn induce a different distribution on the queries asked by *Dec* (this is property (iii) described above). Nevertheless, we are able to show by careful analysis of the distribution of each query in the sequence that the probability of encountering a new linear relation is a polynomial fraction that can be made arbitrarily low. The details of the analysis are given in the next section.

Finally, showing that property (ii) holds is relatively straightforward: we show that after performing sufficiently many encryptions of a random bit, it is unlikely that a new frequently accessed point or a new linear relation will be discovered when the challenge ciphertext is generated. Note that this property is independent from the fake encoding function σ_T and the fake secret key SK_* since encryption is always performed using the actual oracle.

6 Analysis of RIBE adversary

6.1 Hybrid Experiments

In our analysis we compare the view generated by our adversary to several other experiments involving the algorithms of the IBE. For each experiment we show the unlikelihood of certain events which would correspond to bad events for the run of our attack. Relying on the unlikelihood of the bad events, we show that with high probability, our adversary presents a consistent and appropriately distributed view to the decryption algorithm, which allows us to use the correctness property to conclude that our adversary decrypts the challenge ciphertext with high probability. Specifically, the experiments we consider are the following:

Experiment Exp_0

This is the game of IBE security played with our adversary. Our goal is to show that our adversary wins in this experiment with high probability.

Experiment Exp_1

This experiment proceeds as Exp_0 except step 5, which is modified as follows:

Step 5. Let $T^{(*)}_{enc}$ be the set of query-answer pairs that appear during the generation of the challenge ciphertext C_*. Simulate $Dec(\mathrm{SK}_*, C_*)$, where SK_* is the sampled private key for ID_* and C_* is the challenge ciphertext. To answer a query $\mathbf{q} = \mathbf{q^{(1)}} + \mathbf{q^{(g)}} + \mathbf{q^{(t)}}$ let $\hat{\mathbf{q}} = \mathsf{CHB}_{T^{(*)}_{enc}}(\mathbf{q})$. The query $\hat{\mathbf{q}}$ is of the form $\hat{\mathbf{q}} = \hat{\mathbf{q}}^{(\mathbf{1})} + \hat{\mathbf{q}}^{(\mathbf{g})}$. If $\sigma_T(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$ is defined, respond to $\mathbf{q}$ with $\sigma_T(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$. Otherwise, query the actual oracle, and respond with $\mathcal{O}(\hat{\mathbf{q}})$.

Experiment Exp_2

Experiment Exp_2 proceeds as Exp_1, except for the following modification of steps 4 and 5:

Step 4, 5. Let T^*_j be the sets of the first j query-answer pairs that appear during the challenge encryption, and the encryption and decryption procedures in steps 4 and 5. Note that this includes both the sampling of step 4, the generation of the challenge ciphertext, and its decryption in step 5. Specifically, j is in the range $[2n^{c_4+1}]$ (each of the n^{c_4} encryption-decryption processes generates exactly $2n$ queries); note that this is a new use of the notation j.

For each $0 \leq j \leq 2n$ let

$$T^{(j)}_{\mathsf{step1}} \overset{\text{def}}{=} T \cup T^*_j \cup \bigcup_{i=1}^{k_1} (T^{(i)}_{enc} \cup T^{(i)}_{dec})$$

Let $\sigma^{(j)}$ be the minimal partial encoding function[7] that is defined for all mappings $\{\mathbf{q} \mapsto g\} \in T^{(j)}_{\mathsf{step1}}$.

To answer the jth oracle query $\mathbf{q}$ of *Enc* or *Dec*, for $1 \leq j \leq 2n$, respond with $\sigma^{(j)}(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$ if it is defined. Otherwise, respond with $\mathcal{O}(\mathbf{q})$.

Experiment Exp_3

Experiment Exp_3 proceeds as Exp_2, except that step 3 is omitted, and the following modifications are made to steps 4 and 5:

Steps 4, 5. Proceed as in Exp_2, except to answer the jth oracle query $\mathbf{q}$ of *Enc* or *Dec*, for $1 \leq j \leq 2n$ do: respond with $\sigma^{(j)}(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$ if it is defined. Otherwise, choose uniformly at random $g \in_{\mathrm{R}} S \setminus \left\{ g | \exists \mathbf{q} \text{ s.t. } (\mathbf{q}, g) \in T^{(j-1)}_{\mathsf{step1}} \right\}$, and respond to $\mathbf{q}$ with g.

Experiment Exp_4

Experiment Exp_4 is essentially identical to Exp_3, with the following conceptual modification of steps 4 and 5:

Steps 4 and 5. Let $\hat{\sigma}$ be a complete encoding function (i.e. $\hat{\sigma}$ is defined on all of $\mathbb{Z}_p$) that is chosen at random from all encoding functions consistent with the view of the adversary *view* at the last iteration of step 3, and the set T (of query-answer pairs generated by the adversary offline). Let $\hat{\mathcal{O}}$ be the generic group oracle for $\hat{\sigma}$.

Choose $b_* \in_{\mathrm{R}} \{0, 1\}$, and using $\hat{\mathcal{O}}$ generate $C_* \leftarrow_{\mathrm{R}} Enc(\mathrm{PP}, \mathrm{ID}_*, b)$. Then, using $\hat{\mathcal{O}}$ again, compute $Dec(\mathrm{SK}_*, C_*)$.

Experiment Exp_5

The final experiment Exp_5 is identical to Exp_4 except that in steps 4 and 5, the ciphertexts are decrypted using the correct oracle and private key. Specifically, let MSK_0 be the global secret key generated by the challenger, and compute $\mathrm{SK}_* = KeyGen(\mathrm{MSK}_0, \mathrm{ID}_*)$, and $Dec(\mathrm{SK}_*, C_*)$ using the actual oracle $\mathcal{O}$.

[7]Obviously there is a well-defined partial encoding function induced by T. Now, by definition of the sampling (drawn uniformly from a *consistent* view) the $T^{(i)}_{enc}$'s start by computing on the g_i's for which we already have a well-defined discrete logarithm. Similarly, for $T^{(i)}_{dec}$. The issue exists with T^*_j and in particular with the queries made during the computation of the challenge ciphertext by *Enc*. The problem is that there may be a collision, i.e. by accident the actual oracle returns a group element which also appears in some of the other tables. In this case, $\sigma^{(j)}$ may not be well-defined. However, due to step 2 a very similar claim as Claim 2 yields that such an event happens with probability $\leq n/e^n$. This issue affects the comparison of Exp_2, Exp_3, Exp_4, and Exp_5. To simplify the exposition we carry the comparisons conditioned that this even does not happen and we account for it at the end.

6.2 Comparing the intermediate experiments

In this section we present a formal comparison of the above hybrid experiments. Intuitively, our goal is to show that the probability that the adversary decrypts the challenge ciphertext C_* correctly in Exp_0 is close the probability that C_* is decrypted correctly in Exp_4, and to relate the latter probability to the correctness parameter of the IBE.

We start our analysis from the last experiment, establishing properties of Exp_5 first, and proceeding to show that experiments Exp_3, Exp_4, and Exp_5 are essentially identical, and comparing experiments Exp_3 and Exp_2, and finally Exp_2 and Exp_1.

Properties of Exp_5

We make use of the following two properties of Exp_5. First, we observe that the probability that C_* is decrypted correctly in step 5 is exactly the correctness parameter ε of the IBE. Second, we show that it is unlikely that during step 5 we will see any affine relationships among the group elements of the public key (the g_i's) that are not in the span of the relationships that were discovered in step 1.

The following claim follows in a straightforward manner from the fact that in Exp_5 the challenge ciphertext is decrypted using the correct oracle and private key for ID_*.

Claim 3. *Let b be the random variable representing the output of the decryption algorithm in step 5 of Exp_4. Then, $\Pr[b = b_*] = \varepsilon$, where ε is the correctness parameter of $\mathcal{IBE}$.*

We now formally state the second property of experiment Exp_5. The proof is a straightforward adaptation of the proof of Claim 1.

Claim 4. *Let $R = \mathsf{Eq}\left(\bigcup_{i=1}^{l}(T_{enc}^{(i)} \cup T_{dec}^{(i)})\right)$, and let $R_* = \mathsf{Eq}(T_{enc}^{\mathsf{step4}} \cup T_{dec}^{\mathsf{step4}} \cup T_{enc}^{(*)} \cup T_{dec}^{(*)} \cup \bigcup_{i=1}^{l}(T_{enc}^{(i)} \cup T_{dec}^{(i)}))$. Then, $\Pr[R_* \nsubseteq R] \leq 1/n^{c_1 - 1}$*

Comparing Exp_4 and Exp_5

We now show that any (appropriately defined) event is equally likely to occur in experiments Exp_4 and Exp_5. To prove the following claim we also make use of the following simple fact.

Fact 1. *Let Ω be a probability space, and let f be function with domain Ω. Consider the following experiment: 1. sample x from Ω; 2. sample x' from Ω conditioned on $f(x) = f(x')$. Then, for every $y \in \Omega$, $\Pr[x' = y] = \Pr[x = y]$.*

Claim 5. *Let view be the view of the adversary in the last iteration of step 3. Then the tuples $(view, MSK, \hat{\sigma})$ and $(view, MSK_0, \sigma)$ from experiments Exp_4 and Exp_5 respectively are identically distributed.*

Proof. From the description of Exp_4 it is clear that there all oracle queries of Dec will be answered according to the encoding function $\hat{\sigma}$. Therefore, all we need to show is that the global secret key MSK generated in the last iteration of step 3, and the encoding function $\hat{\sigma}$ are uniformly distributed among all strings of length n and injective functions from $\mathbb{Z}_p$ to S.

Let MSK_0 and σ be the actual master secret key and encoding function chosen by the challenger. We show that $(\mathrm{MSK}, \hat{\sigma})$ have the same distribution as (MSK_0, σ). Consider an alternative way of choosing the encoding function σ: first compute *Setup*(MSK$_0$), answering each oracle query with a random element from S (without repetition). Let T_0 be the set of query-answer pairs that appeared during the computation. Then, choose σ uniformly at random from all encoding functions such that for every mapping $\{a\mathbf{1} \mapsto g\} \in T_0$, $\sigma(a) = g$. Clearly this procedure yields the same distribution over (MSK_0, σ) as in the case where we first choose σ randomly and use it to compute *Setup*(MSK$_0$).

Now, notice that *view* is simply a probabilistic (but non-adaptive) function of MSK$_0$, T_0, and σ. We can therefore apply Fact 1 to obtain that $(\mathrm{MSK}, T, \hat{\sigma})$ are distributed identically to $(\mathrm{MSK}_0, T_0, \sigma)$. □

Comparing Exp_3 and Exp_4

The difference between experiments Exp_3 and Exp_4 is purely conceptual: in Exp_3 we essentially generate the needed part of $\hat{\sigma}$ online, as needed, rather than choosing the entire encoding function at once. We therefore have the following property:

Claim 6. *Let view be the view of the adversary in the last iteration of step 3, and let $T_{enc}^{(*)}$ and $T_{dec}^{(*)}$ be the sets of query and answer pairs that appear during the generation of the challenge ciphertext C_* and its decryption. Then, the tuple $(view, MSK, T_{enc}^{(*)}, T_{dec}^{(*)})$ is identically distributed in experiments Exp_3 and Exp_4.*

Comparing Exp_2 and Exp_3

The difference between experiments Exp_2 and Exp_3 is in the way that we answer oracle queries $\mathbf{q}$ for which the response is not determined by the view of the adversary and the set $T_{step1}^{(j)}$. In experiment Exp_3 we generate a response at random from among all the available elements of S, while in Exp_2 the oracle $\mathcal{O}$ is used. The difference is subtle: the response in Exp_3 is what we expect if the view of adversary had been correct. However, in Exp_2 there are three cases: the first case is when there has been a previous query $\mathbf{q}'$ such that $\mathcal{O}(\mathbf{q}) = \mathcal{O}(\mathbf{q}')$ and $\mathbf{q}'$ has not been seen by the adversary, and is not one of the queries asked during the generation of the challenge ciphertext. The only candidates for such queries $\mathbf{q}'$ are oracle queries that were asked by *Setup* or *KeyGen* when the challenger generated the public parameters PP and the private keys $\mathrm{SK}_1, \ldots, \mathrm{SK}_{k_1}$.

The second case is when $\mathbf{q}$ is the first query with discrete logarithm $\mathrm{dlog}_\sigma(\mathbf{q})$ that is submitted to $\mathcal{O}$ in the experiment. Then, the response will be random among all unused elements of S. A minor problem is caused by the fact that the unused elements of S in the view of the adversary, and in the view of $\mathcal{O}$ can be different. However, this difference has a very minor effect on events in the two experiments for large enough sets S (when the order of the group is small any IBE scheme can be broken by querying $\mathcal{O}$ on all points of $\mathbb{Z}_p$).

Finally, we must handle queries that were observed by the adversary in step 3. Note that these queries are not included in $T_{\mathsf{step1}}^{(j)}$, and so in Exp_3 answers to these queries will be generated at random, rather than using the answers that were previously obtained from $\mathcal{O}$. To handle this last issue we explicitly exclude step 3 from the comparison.

In both the following claims we use the following notation:

$$T_{\mathsf{extended}}^{(j)} \stackrel{\mathrm{def}}{=} T \cup T_j^{(*)} \cup \bigcup_{i=1}^{l+k_3} (T_{enc}^{(i)} \cup T_{dec}^{(i)})$$

The following two claims show that case (i) is unlikely to occur. The proof of the first claim is similar to Claim 1 and is omitted.

Claim 7. *Let MSK_0 be the global private key generated by the challenger. Let T_{init} be the set of query-answer pairs that appear during the computation of $Setup(MSK_0)$, and $KeyGen(MSK_0, ID_i)$ for $1 \leq i \leq l$, that are performed by the challenger. Let $R = \mathsf{Eq}\left(T_{init} \cup \bigcup_{l+1 \leq j \leq l+k_3} (T_{enc}^{(j)} \cup T_{dec}^{(j)})\right)$, and let $R_* = \mathsf{Eq}\left(T_{init} \cup T_{enc}^{(*)} \cup T_{dec}^{(*)} \cup \bigcup_{l+1 \leq j \leq l+k_3} (T_{enc}^{(j)} \cup T_{dec}^{(j)})\right)$. Then, in Exp_2, $\Pr[R_* \not\subseteq R] \leq 1/n^{c_3-1}$.*

Claim 8. *For a set $\hat{T}$ of query-answer pairs, let $S(\hat{T}) = \left\{g | \exists \mathbf{q} \text{ s.t. } (\mathbf{q}, g) \in \hat{T}\right\}$. Then, in Exp_2*

$$\Pr\left[S\left(T_{enc}^{(*)} \cup T_{dec}^{(*)}\right) \cap S(T_{init}) \not\subseteq S\left(\bigcup_{l+1 \leq j \leq l+k_3} (T_{enc}^{(j)} \cup T_{dec}^{(j)})\right)\right] \leq 1/n^{c_3-c_1-1}$$

Proof. The claim follows from the fact that the encryption and decryption of the challenge in Exp_2 proceed identically to the online phases of step 3. Therefore, step 5 can be viewed as iteration k_3+1 of step 3, and therefore a random iteration among n^{c_3} iterations of the procedure described in step 3. There are at most n^{c_1+1} elements in $S(T_{init})$. Therefore, the probability that a new one is discovered during the encryption and decryption of the challenge is at most $1/n^{c_3-c_1-1}$. □

The following claim shows that experiments Exp_2 and Exp_3 proceed identically with high probability.

Claim 9. *Let $view_{-3}$ be the view of the adversary in the last iteration of step 2, let $(T_{enc}^{\mathsf{step4}}, T_{dec}^{\mathsf{step4}})$, and let $(T_{enc}^{(*)}, T_{dec}^{(*)})$ be the sets of query and answer pairs that appear during the generation of the challenge ciphertext C_* and its decryption in step 5. Let Γ be the set of all tuples of the form*

$$(view_{-3}, T, MSK, T_{enc}^{\mathsf{step4}}, T_{dec}^{\mathsf{step4}}, T_{enc}^{(*)}, T_{dec}^{(*)})$$

and let D_2 and D_3 be the distributions on Γ induced by experiments Exp_2 and Exp_3 respectively. Then, for all events $E \subseteq \Gamma$, $|\Pr_{\gamma \leftarrow_R D_2}[\gamma \in E] - \Pr_{\gamma \leftarrow_R D_3}[\gamma \in E]| \leq 8n/p + 1/n^{c_3-1} + 1/n^{c_3-c_1-1}$

Proof. We prove the claim through a coupling argument. Consider an alternative way of running experiment Exp_2. Instead of choosing the encoding function σ in advance, generate it one point at a time. Now, consider the following intermediate distributions D_2' and D_3' where each query with a new discrete logarithm is answered using a random element from S (and so collisions may occur). Using a straightforward adaptation of the analysis in Lemma 2, we obtain that

$$|\Pr_{\gamma \leftarrow_R D_2}[\gamma \in E] - \Pr_{\gamma \leftarrow_R D_2'}[\gamma \in E]| \leq 4n/p \quad \text{and}$$
$$|\Pr_{\gamma \leftarrow_R D_3}[\gamma \in E] - \Pr_{\gamma \leftarrow_R D_3'}[\gamma \in E]| \leq 4n/p$$

Therefore, by the triangular inequality it is sufficient to compare the distributions D_2' and D_3'. Consider generating both distributions together in a coupled manner (using identical randomness) up to step 3 of D_2'. Then, proceed to run step 3 of D_2' and let T_{step3} be the set of query-answer pairs (from the actual oracle $\mathcal{O}$) that are recorded. Finally, we run steps 4 and 5 of both D_2' and D_3'

as follows. To answer an oracle query $\mathbf{q}$, let $\hat{\mathbf{q}} = \mathbf{q}$ if $\mathbf{q}$ is a query of Enc, and $\hat{\mathbf{q}} = \mathsf{CHB}_{T_{enc}}(\mathbf{q})$, if $\mathbf{q}$ is a query of Dec, where T_{enc} is the set of query-answer pairs that appeared during the generation of the ciphertext being decrypted.

We consider the following cases:

1. If $\sigma^{(j)}(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$ is defined, then the answer to $\mathbf{q}$ is determined and identical in both experiments.

2. Else, if there exists $\mathbf{q}'$ such that $\{\mathbf{q}' \mapsto \mathcal{O}(\mathbf{q}')\} \in T_{init}$ and $\mathcal{O}(\mathbf{q}) = \mathcal{O}(\mathbf{q}')$ it is answered as follows. If

$$\mathcal{O}(\mathbf{q}) \in S\left(\bigcup_{l+1\le j\le l+k_3} (T_{enc}^{(j)} \cup T_{dec}^{(j)})\right) \qquad \text{and}$$
$$\mathbf{q} - \mathbf{q}' \in \mathrm{span}\left(\mathsf{Eq}\left(\bigcup_{l+1\le j\le l+k_3} (T_{enc}^{(j)} \cup T_{dec}^{(j)})\right)\right)$$

   then in both Exp_2 and Exp_3, the response to $\mathbf{q}$ is $\mathcal{O}(\mathbf{q}')$. Otherwise, if $\mathbf{q}-\mathbf{q}'$ is not in the span, we give up on generating both distributions together, and generate the remaining answers to queries for each of the distributions D_2' and D_3' separately.

3. Finally, the remaining case is when $\mathrm{dlog}_\sigma(\mathbf{q}) \in \mathrm{dlog}_\sigma(T^{(j)}_{\mathsf{extended}})$ but $\sigma^{(j)}(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$ is not defined. In this case, $\mathcal{O}(\mathbf{q})$, which is the answer given to $\mathbf{q}$ in Exp_2 is a value that has already been observed by the adversary, while in Exp_3 a fresh random answer is generated. It remains to show that even though $\mathcal{O}(\mathbf{q})$ is part of the adversary's view (and in particular this value has influenced the choice of T), it is still uniformly distributed given $view_{-3}$.

   Let g be a random variable representing the response to $\mathbf{q}$ in Exp_3. Let $\mathcal{E}$ be the event that $\sigma^{(j)}(\mathrm{dlog}_{\sigma_T}(\mathbf{q}))$ is not defined, and the events in 1 and 2 above do not occur.

$$\Pr[g = \mathcal{O}(\mathbf{q}) | view_{-3},\ \mathcal{E}] = \frac{\Pr[view_{-3} | g = \mathcal{O}(\mathbf{q}),\ \mathcal{E}] \Pr[g = \mathcal{O}(\mathbf{q}) | \mathcal{E}]}{\Pr[view_{-3} | \mathcal{E}]}$$

   We shall now argue that $\Pr[view_{-3} | g = \mathcal{O}(\mathbf{q}),\ \Gamma] = \Pr[view_{-3} | \Gamma]$. This will in turn imply that $view_{-3}$ does not provide any information about $\mathcal{O}(g)$. First, let us argue that $view_{-3}$ does not contain any queries $\mathbf{q}'$ for which $\mathrm{dlog}_\sigma(\mathbf{q}') = \mathrm{dlog}_\sigma(\mathbf{q})$. Otherwise, since $\sigma^{(j)}(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$ is not defined, we would have $\mathbf{q} - \mathbf{q}' \notin \mathrm{span}\left(\mathsf{Eq}\left(\bigcup_{l+1\le j\le l+k_3}(T_{enc}^{(j)} \cup T_{dec}^{(j)})\right)\right)$, which contradicts the conditioning on $\mathcal{E}$.

Clearly, both experiments proceed identically unless there is a query $\mathbf{q}$ that is asked during the encryption or decryption of the challenge ciphertext for which from Claim 7 and Claim 8, we obtain that this occurs with probability at most $1/n^{c_3-1} + 1/n^{c_3-c_1-1}$, which gives us the claim. $\square$

Notational remark: in every experiment in the remaining comparisons, the "view" of the adversary does not include step 3. From this point on we denote by *view* what we defined as $view_{-3}$.

Comparing Exp_1 and Exp_2

The difference between Exp_1 and Exp_2 is in the way that we handle oracle queries $\mathbf{q}$ for which the partial encoding function σ_T does not determine a response (i.e. $\sigma_T(\mathrm{dlog}_{\sigma_T}(\mathbf{q}))$ is not defined). In Exp_1, we forward all such queries to the actual oracle $\mathcal{O}$, whereas in Exp_2 we first check if answer to an equivalent query has already been given, and if so, use that answer instead of querying the oracle.

We start by showing that in Exp_1 it is unlikely that during step 5 new linear relations (under the original oracle encoding function σ) among the group elements g_i of the public key are discovered.

Claim 10. *Let* $R = \mathsf{Eq}\left(\bigcup_{1\le j\le l}(T^{(j)}_{enc} \cup T^{(j)}_{dec}) \cup \bigcup_{l+1\le j\le l+k_3}(T^{(j)}_{enc} \cup T^{(j)}_{dec})\right)$, *and let*

$$R_* = \mathsf{Eq}\left(T^{(*)}_{enc} \cup T^{(*)}_{dec} \cup \bigcup_{1\le j\le l}(T^{(j)}_{enc} \cup T^{(j)}_{dec}) \cup \bigcup_{l+1\le j\le l+k_3}(T^{(j)}_{enc} \cup T^{(j)}_{dec})\right)$$

Then, in Exp_1, $\Pr[R_* \not\subseteq R] \le 1/n^{c_3-1}$; *i.e. with probability* $\ge 1 - n^{c_3-1}$ *these two vector spaces are equal.*

Proof sketch. The proof of this claim is similar to the proof of Claim 1. Essentially, it follows from the fact that in Exp_1, step 5 is identical to an iteration of step 3 where in the online phase oracle queries are answered as in Exp_1 (and the $T^{(i)}_{enc}$ and $T^{(i)}_{dec}$ are constructed). Therefore, step 5 can be viewed as the online phase of another iteration of the procedure described in step 3. Since step 3 is repeated k_3 times, which is a uniformly chosen integer between 1 and n^{c_3}, step 5 can be viewed as a random iteration among n^{c_3} iterations of step 3. R and R_* both contain formal sums of dimension $\le n$. Therefore, at most n out of the n^{c_3} iterations may introduce linearly independent affine relations between the g_i. The probability that this occurs during step 5 is therefore at most $1/n^{c_3-1}$. □

The following claim shows that with high probability our adversary learns all the points of $\mathbb{Z}_p$ (and their corresponding encodings) that are accessed with high probability by the encryption algorithm. The proof is similar to the proof of Claim 2 and is omitted.

Claim 11. *Let* S_δ *be the set of* $x \in \mathbb{Z}_p$ *such that*

$$\Pr\left[x \in \mathrm{dlog}_\sigma(T_{enc})\ ;\ \begin{array}{c} ID \leftarrow_R \mathcal{ID},\ b \leftarrow \{0,1\}, \\ (C, T_{\mathsf{enc}}) \leftarrow_R Enc(PP, ID, b_0) \end{array}\right] \ge \delta$$

Then, $\Pr\left[S_{1/n^{c_2-1}} \subseteq \mathrm{dlog}_\sigma(\bigcup_{i=k_1+1}^{l} T^{(i)}_{enc})\right] \ge 1 - n/e^n$

We are now ready to compare experiments Exp_1 and Exp_2.

Claim 12. *Let* Γ *be the set of all tuples of the form*[8] $(view, T, MSK, T^{(*)}_{enc}, T^{(*)}_{dec})$, *and let* D_1 *and* D_2 *be the distributions on* Γ *induced by experiments* Exp_1 *and* Exp_2 *respectively. Then, for all events* $E \subseteq \Gamma$, $|\Pr_{\gamma\leftarrow_R D_1}[\gamma \in E] - \Pr_{\gamma\leftarrow_R D_2}[\gamma \in E]| \le 4/n^{c_3-1} + 1/n^{c_3-c_1-1} + 1/n^{c_1-1} + 1/n^{c_2-2} + 9n/p$

[8]Recall that $view := view_{-3}$.

Proof. Once again, we apply a coupling argument. Consider running both experiments together using identical randomness. Then, Exp_1 and Exp_2 proceed identically unless one of the following bad events occurs:

Event F_1: during the generation of the challenge ciphertext *Enc* submits a query $\mathbf{q}$ such that $\sigma_T(\mathrm{dlog}_{\sigma_T}(\mathbf{q}))$ is defined and equal to some $\alpha \in S$, but $\mathcal{O}(\mathbf{q}) \neq \alpha$.

Event F_2: For queries $\mathbf{q}_j$ of *Enc* ($1 \leq j \leq n$), define $\hat{\mathbf{q}}_j = \mathbf{q}_j$ (this is for notational convenience; for $n+1 \leq j \leq 2n$, $\hat{\mathbf{q}}_j$ is defined as before: $\hat{\mathbf{q}}_j = \mathsf{CHB}_{T_{enc}^{(*)}}(\mathbf{q}_j)$). Event F_2 is defined as follows: event F_1 does not occur, and during the computation of *Enc* (generation of the challenge ciphertext) or *Dec* a query $\mathbf{q}_j$ is asked for which $\sigma^{(j-1)}(\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}))$ is defined and equal to some $\alpha \in S$, but $\mathcal{O}(\hat{\mathbf{q}}) \neq \alpha$.

Event F_3: during the computation of *Enc* or *Dec* two queries $\mathbf{q}_i$ and $\mathbf{q}_j$, $i < j$, are asked such that $\mathcal{O}(\mathbf{q}_i) = \mathcal{O}(\mathbf{q}_j)$ but $\mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}_i) \neq \mathrm{dlog}_{\sigma_T}(\hat{\mathbf{q}}_j)$.

Let us bound the probability that any of the above events occur.

First, we bound the probability of F_1. Let $X = \{x \in \mathbb{Z}_p | \exists g \in S \text{ s.t. } \sigma_T(x) = g\}$; i.e. X is the domain of the partial encoding function σ_T. Consider a query $\mathbf{q}$ asked by the encryption algorithm. If $\mathrm{dlog}_\sigma(\mathbf{q}) \in S_{1/n^{c_2}-1}$ then by Claim 11 we know that with probability $\geq 1 - n/e^n$ there exists another query $\mathbf{q}' \neq \mathbf{q}$ such that $\mathcal{O}(\mathbf{q}') = \mathcal{O}(\mathbf{q})$ and $\mathbf{q}'$ appeared during an encryption that was simulated by the adversary in step 2. From Claim 10 we know that with probability $\geq 1 - 1/n^{c_3-1}$, $\mathbf{q} - \mathbf{q}' \in R$.

Let $\{c_\mathbf{e} \in \mathbb{Z}_p | \mathbf{e} \in R\}$ be coefficients such that $\mathbf{q} - \mathbf{q}' = \sum_{\mathbf{e}\in R} c_\mathbf{e}\mathbf{e}$. Then,

$$\mathrm{dlog}_{\sigma_T}(\mathbf{q}) = \mathrm{dlog}_{\sigma_T}\left(\mathbf{q}' - \sum_{\mathbf{e}\in R} c_\mathbf{e}\mathbf{e}\right) = \mathrm{dlog}_{\sigma_T}(\mathbf{q}')$$

Therefore, if $\mathrm{dlog}_{\sigma_T}(\mathbf{q}) \in X$ then $\sigma_T(x) = \mathcal{O}(\mathbf{q}') = \mathcal{O}(\mathbf{q})$ (since σ_T is chosen to be consistent with the view of the adversary). Combining the above we get

$$\Pr[F_1 \text{ for an } F_1 \text{ which is caused by } \mathbf{q} \text{ s.t. } \mathrm{dlog}_\sigma(\mathbf{q}) \in S_{1/n^{c_2-1}}] \leq 1/n^{c_3-1} + n/e^n$$

Now, consider all elements $x \in X \setminus S_{1/n^{c_2-1}}$. There are at most n such elements (since $|X| \leq n$). Therefore, by the union bound, the probability that during the generation of the challenge ciphertext a query $\mathbf{q}$ is asked such that $\mathrm{dlog}_\sigma(\mathbf{q}) \in x \in X \setminus S_{1/n^{c_2-1}}$ is at most n/n^{c_2-1}. Consequently we get

$$\Pr[F_1] \leq 1/n^{c_2-2} + 1/n^{c_3-1} + n/e^n$$

Next, we consider the event F_2. Let $R_{\mathsf{step1}} = \mathsf{Eq}\left(\bigcup_{i=1}^{k_1}(T_{enc}^{(i)} \cup T_{dec}^{(i)})\right)$. F_2 can only occur when F_1 does not occur, therefore we can only consider queries $\mathbf{q}_j$ for which $\sigma_T(\mathrm{dlog}_{\sigma_T}(\mathbf{q}_j))$ is not defined. Let $\mathbf{q}'$ be a query for which $\{\mathbf{q}' \mapsto \mathcal{O}(\mathbf{q}_j)\} \in T_{\mathsf{extended}}^{(j-1)}$. Then, if $\hat{\mathbf{q}}_j - \mathbf{q}' \in \mathrm{span}(R_{\mathsf{step1}})$ we have $\mathcal{O}(\hat{\mathbf{q}}_j - \mathbf{q}') = \mathcal{O}(0)$, and therefore $\mathcal{O}(\hat{\mathbf{q}}_j) = \mathcal{O}(\mathbf{q}')$. On the other hand, combining Claims 9, 6, 5 and 4 we get

$$\Pr[F_2] \leq \Pr[\exists j \text{ s.t. } \hat{\mathbf{q}_j} - \mathbf{q}' \notin \mathrm{span}(R_{\mathsf{step1}})] \leq 8n/p + 1/n^{c_3-1} + 1/n^{c_3-c_1-1} + 1/n^{c_1-1}$$

To conclude our analysis we bound the probability of event F_3. If event F_3 occurs, then $\mathbf{q}_i - \mathbf{q}_j \notin R$, but according to Claim 10 this can happen with probability at most $1/n^{c_3-1}$.

Combining the above analysis, we obtain that $\Pr[F_1 \vee F_2 \vee F_3] \leq 2/n^{c_3-1} + 8n/p + 1/n^{c_3-1} + 1/n^{c_3-c_1-1} + 1/n^{c_1-1} + 1/n^{c_2-2} + n/e^n$. $\square$

Comparing Exp_0 and Exp_1

The proof of the following claim follows similarly to the proof of Claim 2 (and 11) and it is omitted. Having three instead of one more general lemma improves readability (note that the individual claims are in principle the same, but the setting is different).

Claim 13. *For $x \in \mathbb{Z}_p$, let E_x be the event that during a decryption of a ciphertext in step 4 of the adversary, a query $\mathbf{q}$ is asked such that $\mathrm{dlog}_{\sigma_T}(\mathsf{CHB}_{T_{enc}}(\mathbf{q})) = x$. Let $X = \{x | \Pr[E_x] \geq 1/n^{c_2} - 1\}$, then $\Pr\left[X \subseteq \mathrm{dlog}_{\sigma_T}(T_{dec}^{\mathsf{step4}})\right] \geq 1 - n/e^n$.*

Claim 14. *Let Γ be the set of all tuples of the form $(view, T, MSK, T_{enc}^{(*)}, T_{dec}^{(*)})$, and let D_0 and D_1 be the distributions on Γ induced by experiments Exp_0 and Exp_1 respectively. Then, for all events $E \subseteq \Gamma$, $|\Pr_{\gamma \leftarrow_R D_0}[\gamma \in E] - \Pr_{\gamma \leftarrow_R D_1}[\gamma \in E]| \leq \frac{2}{n^{c_1-1}} + \frac{2}{n^{c_3-c_1-1}} + \frac{4}{n^{c_3-1}} + \frac{2}{n^{c_2-1}} + \frac{n+1}{e^n} + \frac{16n}{p}$.*

Proof. Let $T_{enc}^{(*)}$ be the set of query-answer pairs asked during the generation of the challenge ciphertext. We couple the two experiments. Clearly, both experiments proceed identically, unless in step 5 the decryption algorithm asks a query $\mathbf{q}$ for which $\sigma_T(\mathsf{CHB}_{T_{enc}^{(*)}}(\mathbf{q}))$ is defined, and yet the adversary responds to $\mathbf{q}$ with $\alpha \neq \sigma_T(\mathsf{CHB}_{T_{enc}^{(*)}}(\mathbf{q}))$.

Let us consider an arbitrary $x \in \mathrm{dlog}_{\sigma_T}(T)$. For each query $\mathbf{q}$ of *Dec*, if $\mathrm{dlog}_{\sigma_T}(\mathsf{CHB}_{T_{enc}^{(*)}}(\mathbf{q})) = x$ we must respond with $\sigma_T(x)$. If $x \in X$ (where X is the set defined in Claim 13), then by Claim 13 we know that with probability $1 - n/e^n$ there exists a pair (mapping) $\{\mathbf{q}' \mapsto \alpha\} \in T_{dec}^{\mathsf{step4}}$ such that $\mathrm{dlog}_{\sigma_T}(\mathbf{q}') = x$. Let $R = \mathsf{Eq}\left(\bigcup_{i=1}^{l}(T_{enc}^{(i)} \cup T_{dec}^{(i)})\right)$. Then, if $\mathbf{q} - \mathbf{q}' \in R$ then there exists a linear combination $\sum_{\mathbf{e} \in R} c_{\mathbf{e}}\mathbf{e}$ such that $\mathbf{q} = \mathbf{q}' - \sum_{\mathbf{e} \in R} c_{\mathbf{e}}\mathbf{e}$, and $\mathcal{O}(\mathbf{q}) = \mathcal{O}(\mathbf{q}' - \sum_{\mathbf{e} \in R} c_{\mathbf{e}}\mathbf{e}) = \mathcal{O}(\mathbf{q}')$. In this case, our adversary responds to $\mathbf{q}$ with $\sigma_T(x)$. By combining Claims 12, 9, 6, 5 and 4 we get that

$$\Pr[\mathbf{q} - \mathbf{q}' \notin R] \leq \frac{2}{n^{c_1-1}} + \frac{2}{n^{c_3-c_1-1}} + \frac{4}{n^{c_3-1}} + \frac{2}{n^{c_2-1}} + \frac{n}{e^n} + \frac{16n}{p}$$

Now, consider arbitrary $x \notin X$. There are at most $2n$ elements $x \in \mathrm{dlog}_{\sigma_T}(T)$. Therefore, by the union bound the probability that *Dec* asks a query $\mathbf{q}$ for which $\mathrm{dlog}_{\sigma_T}(\mathbf{q}) = x$ is at most $1/n^{c_2-2}$. □

Acknowledgments

This work is in part a result of a delightful collaboration with Dan Boneh and Brent Waters. We'd like to thank Brent for the insightful remarks and suggestions, and Dan for his collaboration in the beginning of this research.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In *CRYPTO*, pages 98–115, 2010.

[ADVW12] S. Agrawal, Y. Dodis, V. Vaikuntanathan, and D. Wichs. On continual leakage of discrete log representations. *IACR Cryptology ePrint Archive*, 2012:367, 2012. informal publication.

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pages 443–459, 2004.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO*, pages 213–229, 2001.

[BF03] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003.

[Bon98] D. Boneh. The Decision Diffie-Hellman problem. In *Third Algorithmic Number Theory Symp*, pages 48–63, 1998.

[BPR$^+$08] D. Boneh, P. A. Papakonstantinou, C. W. Rackoff, Y. Vahlis, and B. Waters. On the impossibility of identity based encryption from trapdoor permutations. In *Foundations of Computer Science (FOCS)*, pages 283–292, 2008.

[CDK$^+$12] R. Cramer, I. Damgård, E. Kiltz, S. Zakarias, and A. Zottarel. DDH-like assumptions based on extension rings. In *PKC*, pages 644–661, 2012.

[CHK03] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *IMA International Conference on Cryptography and Coding*, pages 26–8, 2001.

[CS98] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[DHT12] Y. Dodis, I. Haitner, and A. Tentes. On the instantiability of hash-and-sign rsa signatures. In *Theory of Cryptography Conference (TCC)*, pages 112–132, 2012.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Symposium on the Theory of Computing (STOC)*, pages 169–178, 2009.

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.

[GKM$^+$00] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *Foundations of Computer Science (FOCS)*, pages 325–335, 2000.

[GMM07] Y. Gertner, T. Malkin, and S. Myers. Towards a separation of semantic and CCA security for public key encryption. In *Theory of Cryptography Conference (TCC)*, pages 434–455, 2007.

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *Foundations of Computer Science (FOCS)*, pages 126–135, 2001.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Symposium on the Theory of Computing (STOC)*, pages 197–206, 2008.

[HH09] I. Haitner and T. Holenstein. On the (im)possibility of key dependent encryption. In *Theory of Cryptography Conference (TCC)*, pages 202–219, 2009.

[HHRS07] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *Foundations of Computer Science (FOCS)*, pages 669–679, 2007.

[IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Symposium on the Theory of Computing (STOC)*, pages 44–61, 1989.

[KM07] N. Koblitz and A. Menezes. Another look at generic groups. *Advances in Mathematics of Communications*, 1:13–28, 2007.

[Lan87] Serge Lang. *Linear Algebra.* Springer-Verlag, 1987.

[Mau05] U. M. Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding*, pages 1–12, 2005.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[Pap10] P. A. Papakonstantinou. *Constructions, lower bounds, and new directions in Cryptography and Computational Complexity.* PhD thesis, University of Toronto, March 2010. (pp 43–60) http://itcs.tsinghua.edu.cn/∼papakons/pdfs/phd_thesis.pdf.

[RTV04] O. Reingold, L. Trevisan, and S. Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography Conference (TCC)*, pages 1–20, 2004.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[Sho97] V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, pages 256–266, 1997.

[Sim98] D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *EUROCRYPT*, pages 334–345, 1998.

[Vah10] Y. Vahlis. Two is a crowd? a black-box separation of one-wayness and security under correlated inputs. In *Theory of Cryptography Conference (TCC)*, pages 165–182, 2010.

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, 2005.