Circular and KDM Security for Identity-Based Encryption

Jacob Alperin-Sheriff* Chris Peikert†

March 22, 2012

Abstract

We initiate the study of security for key-dependent messages (KDM), sometimes also known as "circular" or "clique" security, in the setting of identity-based encryption (IBE). Circular/KDM security requires that ciphertexts preserve secrecy even when they encrypt messages that may depend on the secret keys, and arises in natural usage scenarios for IBE.

We construct an IBE system that is circular secure for affine functions of users' secret keys, based on the learning with errors (LWE) problem (and hence on worst-case lattice problems). The scheme is secure in the standard model, under a natural extension of a selective-identity attack. Our three main technical contributions are (1) showing the circular/KDM-security of a "dual"-style LWE public-key cryptosystem, (2) proving the hardness of a version of the "extended LWE" problem due to O'Neill, Peikert and Waters (CRYPTO'11), and (3) building an IBE scheme around the dual-style system using a novel lattice-based "all-but-d" trapdoor function.

1 Introduction

Traditional notions of secure encryption, starting with semantic (or IND-CPA) security [GM82], assume that the plaintext messages do not depend on the secret decryption key (except perhaps indirectly, via the public key or other ciphertexts). In many settings, this may fail to be the case. One obvious scenario is, of course, careless or improper key management: for example, some disk encryption systems end up encrypting the symmetric secret key itself (or a derivative) and storing it on disk. However, there are also situations in which key-dependent messages are used as an integral part of an cryptosystem. For example, in their anonymous credential system, Camenisch and Lysyanskaya [CL01] use a cycle of key-dependent messages to discourage users from delegating their secret keys. More recently, Gentry's "bootstrapping" technique for obtaining a fully homomorphic cryptosystem [Gen09] encrypts a secret key under the corresponding public key in order to support unbounded homomorphism; the cryptosystem therefore needs to be "circular secure." More generally, a system that potentially reveals encryptions of any party's secret key under any user's public key needs to be "clique" or "key-dependent message" (KDM) secure. This notion allows for proving formal symbolic soundness of cryptosystems having complexity-based security proofs [ABHS05].

Since Boneh *et al.*'s breakthrough work [BHHO08] giving a KDM-secure encryption scheme, in the standard model, from the Decision Diffie-Hellman assumption, a large number of results (mostly positive)

*School of Computer Science, College of Computing, Georgia Institute of Technology. Email: jmas6@cc.gatech.edu

†School of Computer Science, Georgia Institute of Technology. Email: cpeikert@cc.gatech.edu. This material is based upon work supported by the National Science Foundation under Grant CNS-0716786 and CAREER Award CCF-1054495, and by the Alfred P. Sloan Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Sloan Foundation.

have been obtained regarding circular- and KDM-secure encryption [HH09, ACPS09, BHHI10, BG10, App11, MTY11, BGK11, BV11]. However, all these works have dealt only with the symmetric or public-key settings; in particular, the question of circular/KDM security for *identity-based* cryptography has not yet been considered. Recall that in identity-based encryption [Sha84], any string can serve as a public key, and the corresponding secret keys are generated and administered by a trusted Private Key Generator (PKG).

Circular security for IBE. In this work we define and construct a circular/KDM-secure identity-based encryption (IBE) scheme. KDM security is well-motivated by some natural usage scenarios for IBE, as we now explain.

Recall that identity-based encryption gives a natural and lightweight solution to revocation, via expiring keys. The lifetime of the cryptosystem is divided into time periods, or "epochs." An identity string consists of a user's "true" identity (e.g., name) concatenated with an epoch; when encrypting, one uses the identity for the current epoch. To support revocation, the PKG gives out a user's secret key only for the current epoch, and only if the user is still authorized to be part of the system. Therefore, a user's privileges can be revoked by simply refusing to give out his secret key in future epochs; in particular, this revocation is transparent to the encrypter.

The above framework makes it necessary for users to periodically get new secret keys from the PKG, confidentially. The most lightweight method, which eliminates the need for the user to prove his identity every time, is simply for the PKG to encrypt the new secret key under the user's identity for the previous epoch. This can be proved secure, assuming the underlying IBE is CPA-secure, *as long as there are no cycles of encrypted keys*. However, if a user deletes or loses an old secret key and wants to decrypt a ciphertext from the corresponding epoch, it would be natural for the authority to provide the old secret key encrypted under the user's identity for the current epoch. But because the current secret key has also been encrypted (perhaps via a chain of encryptions) under the old identity, this may be unsafe unless the IBE is KDM-secure.

1.1 Our Contributions

As already mentioned, in this work we define a form of circular/KDM security for identity-based encryption, and give a standard-model construction based on the learning with errors (LWE) problem, hence on worst-case lattice problems via the reductions of [Reg05, Pei09].

As in prior positive results on circular security [BHHO08, ACPS09, BG10], our definition allows the adversary to obtain encrypted "key cliques" for affine functions of the secret keys. More precisely, for any tuple of identities $(id_1, \ldots, id_d)$, the attacker may adaptively query encryptions of $f(sk_{id_i})$ under any of the identities id_j, where f is any affine function over the message space, and each sk_{id_i} is a secret key for identity id_i. Our attack model is in the style of a "selective identity" attack, wherein the adversary must declare the target identities $id_1, \ldots, id_d$ (but not the functions f) before seeing the public parameters of the scheme. While this is not the strongest security notion we might hope for, it appears to at least capture the main security requirements of the scenarios described above, because encrypted key cycles are only ever published for the same "real-world" identity at different time epochs. Therefore, just as in a standard selective-identity attack for IBE, the adversary is actually limited to attacking just a single real-world identity, on a set of d epochs (which could, for example, include all valid epochs). We also point out that by a routine hybrid argument, security also holds when attacking a *disjoint* collection of identity cliques (that are named before seeing the public parameters).

Our IBE construction is built from two components, both of which involve some novel techniques. First, we give an LWE-based *public-key* cryptosystem that is clique secure (even for an *unbounded* number of users) for affine functions, and is suitable for embedding into an IBE like the one of [GPV08]. Second, we construct a lattice-based "all-but-d" trapdoor function that serves as the main foundation of the IBE. We elaborate on these two contributions next.

Clique-secure public-key cryptosystem. We first recall that Applebaum *et al.* [ACPS09] showed that a variant of Regev's so-called "primal" LWE cryptosystem [Reg05] is clique secure for affine functions. Unfortunately, this primal-type system does not seem suitable as the foundation for identity-based encryption using the tools of [GPV08]. The first reason is that the proof of clique security from [ACPS09] needs the users' public keys to be completely independent, rather than incorporating a shared random string (e.g., the public parameters in an IBE system). The second reason is a bit more technical, and is already described in [GPV08]: in primal-style systems, the user-specific public keys are exponentially sparse pseudorandom values (with unique secret keys), and it is difficult to design an appropriate mapping from identities to valid public keys that actually admit usable underlying secret keys.

Therefore, we first need to obtain clique security for a so-called "dual"-type cryptosystem (using the terminology from [GPV08]), in which *every* syntactically valid public key has a functional underlying secret key (actually, many such secret keys) that can be extracted by the PKG. It turns out that achieving this goal is quite a bit more technically challenging than it was for the "primal"-style schemes. This is primarily because the KDM-secure scheme from [ACPS09] (like the earlier one from [BHHO08]) has the nice property that given the public key alone, one can efficiently generate *statistically well-distributed* encryptions of the secret key (without knowing the corresponding encryption randomness). This immediately implies circular security for "self-loops," and clique security follows from a couple of other related techniques.

Unfortunately, this nice statistical property on ciphertexts does not seem attainable for dual-style LWE encryption, because now valid ciphertexts are exponentially sparse and hard to generate without knowing the underlying encryption randomness. In addition, because the adversary may obtain an *unbounded* number of key-dependent ciphertexts, we also cannot rely on any statistical entropy of the secret key conditioned on the public key, as is common in the security proofs of most dual-style cryptosystems.

We resolve the above issues by relying on computational assumptions twice in our security proof, first when changing the way that challenge ciphertexts are produced (i.e., by using knowledge of the secret key), and then again when changing the form of the public key. Notably, unlike prior works (e.g., [DGK$^+$10, OPW11]) in which ciphertexts in intermediate games are created by "encrypting with an (information theoretically revealed) secret key," we are able to avoid the use of super-polynomially large noise to "overwhelm" the slight statistical difference between the two ways of generating ciphertexts. This lets us prove security under fully polynomial lattice/LWE assumptions, i.e., those involving a polynomially bounded modulus q and inverse error rate for the LWE problem, and therefore polynomial approximation factors for worst-case lattice problems. We do so by proving the hardness of a version of the *extended*-LWE problem, as defined and left open by the recent work of [OPW11]. We believe that this result should be useful in several other contexts as well.

All-but-d trapdoor functions. We use the clique-secure cryptosystem described above as the foundation for a clique-secure IBE. To make the cryptosystem identity-based, as in [GPV08] we need to embed a "strong" trapdoor into the public parameters so that the PKG can extract a secret key for any identity. Here we use the ideas behind the tag-based algebraic construction of [ABB10], and follow the somewhat simpler and more efficient realization recently due to [MP12]. We remark that these trapdoor constructions are well-suited to security definitions in which the adversary attacks a *single* tag, because the trapdoor can be "punctured" (i.e., made useless for extracting secret keys, and useful for embedding an LWE challenge) at exactly one predetermined tag. Unfortunately, this does not appear to be sufficient for our purposes, because in the clique security game, the adversary is attacking d identities at once and can obtain challenge ciphertexts under all of them.

To resolve the insufficiency of a single puncture, we extend the trapdoor constructions of [ABB10, MP12] so that it is possible to puncture the trapdoor at up to d arbitrary, prespecified tags. To accomplish this, we show how to statistically hide in the public key a degree-d polynomial $f(\cdot)$ over a certain ring $\mathcal{R}$, so that $f(id_i) = 0$ for all the targeted tags (identities) id_i, while $f(id)$ is a unit in $\mathcal{R}$ (i.e., is invertible) for all other identities. The d components of the public key can be combined so as to homomorphically evaluate f on any desired tag. The underlying trapdoor is punctured exactly on tags id where $f(id) = 0$, and is effective for inversion whenever $f(id)$ is a unit in $\mathcal{R}$. Our construction is analogous to the one of [CS06] in the setting of prime-order groups with bilinear pairings (where arithmetic "in the exponent" happens in a field), and the all-but-d lossy trapdoor functions of [HLOV11]. However, since lattice-based constructions do not work with fields or rings like $\mathbb{Z}_N$, we instead use techniques from the literature on secret sharing over groups and modules, e.g., [DF94, Feh98].

We remark that, for technical reasons relating to the number of "hints" for which we can prove the hardness of the extended-LWE problem, we have not been able to prove the KDM-security of our IBE under fully polynomial assumptions (as we were able to do for our basic public-key cryptosystem). We instead rely on the conjectured hardness of LWE for a slightly super-polynomial modulus q and inverse error rate $1/\alpha$, which translates via known reductions [Reg05, Pei09] to the conjectured hardness of worst-case lattice problems for slightly super-polynomial approximation factors, against slightly super-polynomial-time algorithms. Known lattice algorithms are very far from disproving such conjectures.

2 Preliminaries

We denote the real numbers by $\mathbb{R}$ and the integers by $\mathbb{Z}$. For a positive integer d, we use $[d]$ to denote the set $\{1, \ldots, d\}$. We denote vectors over $\mathbb{R}$ and $\mathbb{Z}$ with lower-case bold letters (e.g. $\mathbf{x}$), and matrices by upper-case bold letters (e.g. $\mathbf{A}$). We say that a function is *negligible*, written $\mathrm{negl}(n)$, if it vanishes faster than the inverse of any polynomial in n. The *statistical distance* between two distributions X, Y over a finite or countable set D is $\Delta(X, Y) = \frac{1}{2}\sum_{w \in D} |X(w) - Y(w)|$. Statistical distance is a metric, and in particular obeys the triangle inequality. Let $\{X_n\}$ and $\{Y_n\}$ be ensembles of random variables indexed by the security parameter n. We say that X and Y are *statistically close* if $\Delta(X_n, Y_n) = \mathrm{negl}(n)$. For a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$, the *largest singular value* (also known as the *spectral norm*) of $\mathbf{X}$ is defined as $s_1(\mathbf{X}) = \max_{\|\mathbf{u}\|=1} \|\mathbf{X}\mathbf{u}\|$.

2.1 Lattices and Gaussians

A (full-rank) m-dimensional *integer lattice* Λ is an additive subgroup of $\mathbb{Z}^m$ with finite index. This work is concerned with the family of integer lattices whose cryptographic importance was first demonstrated by Ajtai [Ajt96]. For integers $n \geq 1$, modulus $q \geq 2$, an m-dimensional lattice from this family is specified by an "arity check" matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$:

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}^m.$$

For any $\mathbf{y}$ in the subgroup of $\mathbb{Z}_q^n$ generated by the columns of $\mathbf{A}$, we also define the coset

$$\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{y} \bmod q\} = \Lambda^{\perp}(\mathbf{A}) + \bar{\mathbf{x}},$$

where $\bar{\mathbf{x}} \in \mathbb{Z}^m$ is an arbitrary solution to $\mathbf{A}\bar{\mathbf{x}} = \mathbf{y}$.

We briefly recall Gaussian distributions over lattices (for more details see [MR04, GPV08]). For $s > 0$ and dimension $m \geq 1$, the Gaussian function $\rho_s : \mathbb{R}^m \to (0, 1]$ is defined as $\rho_s(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/s^2)$. For a coset $\Lambda + \mathbf{c}$ of a lattice Λ, the *discrete Gaussian distribution* $D_{\Lambda+\mathbf{c},s}$ (centered at zero) assigns probability proportional to $\rho_s(\mathbf{x})$ to each vector in the coset, and probability zero elsewhere.

We will need a few standard concepts and facts about discrete Gaussians over lattices. First, for $\epsilon > 0$ the *smoothing parameter* [MR04] $\eta_\epsilon(\Lambda)$ of an n-dimensional lattice is a positive real value. We will not need its precise definition, which depends on the notion of the *dual* lattice, but only recall the few relevant facts that we need; for details, see, e.g., [MR04, GPV08, MP12].

Lemma 2.1. *Let $m \geq Cn \lg q$ for some constant $C > 1$.*

1. *For any $\omega(\sqrt{\log n})$ function, we have $\eta_\epsilon(\mathbb{Z}^n) \leq \omega(\sqrt{\log n})$ for some negligible $\epsilon(n) = \mathrm{negl}(n)$.*
2. *With all but $\mathrm{negl}(n)$ probability over the uniformly random choice of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the following holds: For $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ where $r = \omega(\sqrt{\log n})$, the distribution of $\mathbf{y} = \mathbf{A}\mathbf{e} \bmod q$ is within $\mathrm{negl}(n)$ statistical distance of uniform, and the conditional distribution of $\mathbf{e}$ given $\mathbf{y}$ is $D_{\Lambda_{\mathbf{y}}^\perp(\mathbf{A}),r}$.*
3. *For any m-dimensional lattice Λ, any $\mathbf{c} \in \mathbb{Z}^m$, and any $r \geq \eta_\epsilon(\Lambda)$ where $\epsilon(n) = \mathrm{negl}(n)$, we have $\|D_{\Lambda+\mathbf{c},r}\| \leq r\sqrt{m}$ with all but $\mathrm{negl}(n)$ probability. In addition, for $\Lambda = \mathbb{Z}$ we have $|D_{\mathbb{Z},r}| \leq r \cdot \omega(\sqrt{\log n})$ except with $\mathrm{negl}(n)$ probability.*
4. *For any $r > 0$, and for $\mathbf{R} \leftarrow D_{\mathbb{Z},r}^{n\times k}$, we have $s_1(\mathbf{R}) \leq r \cdot O(\sqrt{n} + \sqrt{k})$ except with $\mathrm{negl}(n)$ probability.*

Lemma 2.2. *For any real number $r = \omega(\sqrt{\log n})$ and $c \in \mathbb{Z}$, the statistical distance between $D_{\mathbb{Z},r}$ and $c + D_{\mathbb{Z},r}$ is $O(|c|/r)$.*

2.2 Trapdoors for Lattices

We recall the efficient trapdoor construction and associated sampling algorithm of Micciancio and Peikert [MP12]. This construction uses a universal public "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times w}$ for which there is an efficient discrete Gaussian sampling algorithm for any parameter $r \geq \omega(\sqrt{\log n}) \geq \eta_\epsilon(\Lambda^\perp(\mathbf{G}))$ (for some $\epsilon(n) = \mathrm{negl}(n)$), i.e., an algorithm that, given any $\mathbf{y} \in \mathbb{Z}_q^n$ and r, outputs a sample from $D_{\Lambda_{\mathbf{y}}^\perp(\mathbf{G}),r}$. For concreteness, as in [MP12] we take $\mathbf{G} = \mathbf{I}_n \otimes [1, 2, 4, \ldots, 2^{k-1}] \in \mathbb{Z}_q^{n\times nk}$ for $k = \lceil \lg q \rceil$.

Following [MP12], we say that an integer matrix $\mathbf{R} \in \mathbb{Z}^{(m-n)\times w}$ is a "strong" trapdoor with tag H for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ if $\mathbf{A}\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}\end{smallmatrix}\right] = H(\mathbf{G})$ for some efficiently computable and invertible linear transformation H over $\mathbb{Z}_q^n$, which is applied column-wise to $\mathbf{G}$. Equivalently, in place of $H(\mathbf{G})$ we may write $\mathbf{H} \cdot \mathbf{G}$ for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, but in our constructions it will be more natural to work with invertible linear transformations, without explicitly referring to the matrices that represent them.

Lemma 2.3 ([MP12, Theorem 5.1])**.** *Let $\mathbf{R}$ be a strong trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. There is an efficient randomized algorithm that, given $\mathbf{R}$, any $\mathbf{u} \in \mathbb{Z}_q^n$, and any $r \geq s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n}) \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ (for some $\epsilon(n) = \mathrm{negl}(n)$), samples from a distribution within $\mathrm{negl}(n)$ distance of $D_{\Lambda_{\mathbf{u}}^\perp(\mathbf{A}),r}$.*

2.3 Learning With Errors

The *learning with errors* (LWE) problem is parameterized by a dimension $n \geq 1$, an integer modulus $q \geq 2$ and an error distribution χ over $\mathbb{Z}$ (or its induced distribution over $\mathbb{Z}_q$). For a vector $\mathbf{s} \in \mathbb{Z}_q^n$, the distribution

$A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + x)$, where $x \leftarrow \chi$.

The search version of LWE is to recover an arbitrary s given oracle access to $A_{\mathbf{s},\chi}$. The decision version of LWE is to distinguish, with non-negligible advantage, between samples from $A_{\mathbf{s},\chi}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ and uniformly random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. There are search-to-decision reductions for LWE for a variety of moduli q and parameter conditions ([Reg05, Pei09, ACPS09, MM11, MP12]). Of particular importance to us are the reductions from [ACPS09, MP12] for $q = p^e$, where p is prime, $e \geq 1$ is an integer, and $\Pr_{\mathbf{x} \leftarrow \chi}[|\mathbf{x}| \geq p/2] = \text{negl}(n)$. The reductions runs in time polynomial in n, p, and e.

For error distribution $\chi = D_{\mathbb{Z},\alpha q}$, where $\alpha q \geq 2\sqrt{n}$, the search version of LWE is at least as hard as *quantumly* approximating certain worst-case problems on n-dimensional lattices to within $\tilde{O}(n/a)$ factors [Reg05]; for certain parameters, a classical reduction is known for a subset of these lattice problems [Pei09]. Note that the original hardness result for search-LWE was for a continuous Gaussian error distribution, but this can be converted to a discrete Gaussian disribution with a suitable randomized rounding method [Pei10].

We will need the transformation of Applebaum *et al.* [ACPS09] from the standard decision-LWE problem (where s is uniform) to one where the secret s is chosen from the error distribution χ.

Lemma 2.4 ([ACPS09, Lemma 2])**.** *Let $q = p^e$ be a prime power. There is a deterministic polynomial-time transformation that, for arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$ and error distribution χ, maps $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{x}},\chi}$ where $\bar{\mathbf{x}} \leftarrow \chi^n$, and maps $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ to itself. The transformation also produces an invertible square matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times n}$ and $\bar{\mathbf{b}} \in \mathbb{Z}_q^n$ that, when mapping $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{x}},\chi}$, satisfy $\bar{\mathbf{x}} = -\bar{\mathbf{A}}^t\mathbf{s} + \bar{\mathbf{b}}$.*

2.4 Identity-Based Encryption

As usual, an identity-based encryption scheme [BF01] consists of four algorithms: $\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec}$. In our scheme, in addition to the security parameter 1^n, Setup also takes in a parameter d denoting the maximum number of users in a clique . Setup outputs the master public key MPK (which includes the system parameters) and the master secret key MSK. Ext takes in an identity id MPK, MSK and outputs a secret key SK_{id} for identity id. Enc takes in MPK, id and a message μ, and returns a ciphertext c. Dec takes in $\mathsf{MPK}, \mathsf{SK}_{id}$ and ciphertext c, and returns message μ. As usual for lattice-based schemes, correctness requires that with overwhelming probability over the random coins used by the algorithms, we have that for $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^n, d)$, $\mathsf{SK}_{id} \leftarrow \mathsf{Ext}(\mathsf{MPK}, \mathsf{MSK}, id)$ and any message μ: $\mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{id}, \mathsf{Enc}(\mathsf{MPK}, id, \mu)) = \mu$.

2.5 Key-Dependent Message Security

We now proceed to formally define key-dependent message security for public-key encryption and for identity-based encryption. We adapt the original definitions of Black *et al.* [BRS02]. In their original definitions, the adversary plays a game with a challenger, and is able to make encryption queries for functions of the users' secret keys. The adversary is restricted to functions from a certain family $\mathcal{F} \subset \{f : \mathcal{K}^\ell \to \mathcal{M}\}$, where $\mathcal{K}$ is the keyspace for identity secret keys and $\mathcal{M}$ is the message space of the encryption scheme. (Technically, $\mathcal{F}$ is a family of sets of functions parameterized by the security parameter n and the number of users d.) The adversary's goal is to distinguish between honest encryptions of the queried function applied to the secret keys, and encryptions of a fixed dummy value (say, 0).

To simplify our security proofs, in our definition the adversary specifies two functions $(f_0, f_1) \in \mathcal{F}$ with each query, and must distinguish between encryptions of f_0 and encryptions of f_1. If $f(k_1, \ldots, k_d) = 0$ is contained in $\mathcal{F}$ (which should be the case if we want KDM security to imply standard semantic security), then it is easy to see that this definition is at least as strong as (and is in fact equivalent to) the original.

To define KDM-security for identity-based encryption, we extend the definition of selective security for IBE from [CHK03, BCHK07]. An adversary plays a game with a challenger that answer encryption queries for functions of the secret keys for identities from a list $\mathcal{I}$, encrypted under identities from $\mathcal{I}$. For selective security, $\mathcal{I}$ must be specified before the adversary sees the public key and remains static throughout the game. In addition to (key-dependent) encryption queries, the adversary is also allowed to make extraction queries for any identity $id \notin \mathcal{I}$.

Our definitions can be extended to adaptive security as well. In this case, the adversary can adaptively add identities to $\mathcal{I}$ during the course of the game. In order to make the definition meaningful, the adversary is only allowed to add identities to $\mathcal{I}$ for which it has not previously made an extraction query.

For an identity-based encryption scheme $(\mathsf{Setup}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$, the security game between an adversary and a challenger is parameterized by some $\beta \in \{0, 1\}$ and proceeds as follows.

1. $\mathcal{A}(1^n, d)$ outputs a list of (distinct) target identities $\mathcal{I} = (id_1, id_2, \dots id_\ell)$ for some $\ell \leq d$.
2. The challenger runs $(mpk, msk) \leftarrow \mathsf{Setup}(1^n, d)$. The adversary is given mpk. The challenger then extracts secret keys for each of the target identities, running $sk_i \leftarrow \mathsf{Ext}_{msk}(id_i)$ for each $i \in [\ell]$.
3. $\mathcal{A}$ then can make extraction and encryption queries, in the order of its choice.

   Extraction Queries: $\mathcal{A}$ can query $\mathsf{Ext}_{msk}(\cdot)$ for any identity $id \notin \mathcal{I}$

   Encryption Queries: $\mathcal{A}$ can make encryption queries of the form (f_0, f_1, i), where $f_0, f_1 \in \mathcal{F}$ and $1 \leq i \leq \ell$. The challenger computes $m \leftarrow f_\beta(sk_1, \dots, sk_\ell)$ and $c \leftarrow \mathsf{Enc}(id_i, m)$, and returns c to $\mathcal{A}$.

We say that the scheme is selective-identity KDM-CPA secure with respect to $\mathcal{F}$ if the games for $\beta = 0, 1$ are computationally indistinguishable.

We define KDM-CPA security for a public-key scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ in a similar manner. Starting at phase two above (since there are no identities to target), the challenger now runs Gen d times, and gives $pk_1, \dots, pk_d$ to the adversary. In phase three, the adversary can only make encryption queries (since there are no identities to extract), and requests encryptions under public keys instead of identities. Everything else is exactly the same.

3 Hardness of Extended LWE

In this section we describe the *extended*-LWE problem (as originally defined in [OPW11]), and give a reduction to it from the standard LWE problem (with polynomially bounded parameters), thus establishing its hardness under a mild assumption.

3.1 Background and the Problem

O'Neill, Peikert and Waters [OPW11] introduced the extended-LWE problem as a simplifying tool for certain security proofs in which LWE is used in a "hash proof-like" fashion, and additional information about the secret key is revealed to the attacker. In prior works, dealing with such situations often involved adding some "overwhelming" (super-polynomial) extra noise in order to disguise a small but noticeable statistical difference between, e.g., creating a ciphertext honestly according to an encryption algorithm, and creating one by combining the secret key with a challenge LWE instance. Unfortunately, the use of such overwhelming noise requires an underlying LWE problem with super-polynomial modulus q and inverse error rate $1/\alpha$, which corresponds to a substantially stronger assumption than is needed in the security proofs for many other cryptosystems.

Here we recall the formal definition of the extended-LWE problem. In addition to the usual n, q, and χ parameters for LWE, we also have a number $m = \mathrm{poly}(n)$ of LWE samples, an efficiently sampleable "hint" distribution τ over $\mathbb{Z}^m$ (often, a discrete Gaussian $D_{\mathbb{Z},r}^m$ for some $r \geq 1$) and another Gaussian parameter $\beta > 0$. The problem is to distinguish, with non-negligible advantage, between the two experiments described next; the extended-LWE assumption is that this distinguishing problem is hard. In the ExptLWE experiment, the challenger chooses $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, a secret $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and error vector $\mathbf{x} \leftarrow \chi^m$ defining $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{x}^t$, along with a "hint" vector $\mathbf{z} \leftarrow \tau$ and error term $\tilde{x} \leftarrow D_{\mathbb{Z},\beta q}$, and outputs

$$(\mathbf{A}, \mathbf{b}, \mathbf{z}, b' = \langle \mathbf{x}, \mathbf{z} \rangle + \tilde{x}).$$

The first two components are just m LWE samples, while the latter two components may be seen as a hint about the error vector $\mathbf{x} \in \mathbb{Z}^m$ in the form of a (noisy) inner product with a vector $\mathbf{z} \in \mathbb{Z}^m$, which is not reduced modulo anything. The ExptUnif experiment is the same, except that $\mathbf{b}$ is defined to be uniformly random and independent of everything else.

Notice that because $\mathbf{A}$ and $\mathbf{z}$ are public, one can "amortize" the extended-LWE problem by including any $\mathrm{poly}(n)$ number of vectors $\mathbf{b}_i^t = \mathbf{s}_i^t\mathbf{A} + \mathbf{x}_i^t$ and hints $b_i' = \langle \mathbf{x}_i, \mathbf{z} \rangle$, for independent $\mathbf{s}_i, \mathbf{x}_i$ (and the same $\mathbf{A}, \mathbf{z}$). By a routine hybrid argument, the two forms of the problem are equivalent, up to a $\mathrm{poly}(n)$ factor in the distinguishing advantage. We use the amortized form of the problem in our security proof in Section 4.

As observed in [OPW11] (and implicitly in prior works like [GKPV10, DGK$^+$10]), there is a straightforward reduction from LWE with $\chi = D_{\mathbb{Z},\alpha q}$ to extended-LWE where τ is any m-fold product distribution with variance r^2, if the ratio $\beta/(r \cdot \alpha)$ is superpolynomial in n. In fact, in this case we can securely give out an *unbounded* polynomial number of hints $\mathbf{z}_i, b_i' = \langle \mathbf{x}, \mathbf{z}_i \rangle + \tilde{x}_i$ about the error $\mathbf{x}$. This is simply because by Lemma 2.2, the terms $\tilde{x} \leftarrow D_{\mathbb{Z},\beta q}$ statistically hide the inner product $\langle \mathbf{x}, \mathbf{z} \rangle$, since the latter has magnitude $\approx r\|\mathbf{x}\| \leq r\alpha q\sqrt{m} = \beta q \cdot \mathrm{negl}(n)$. As a result, the reduction can just simulate the hints $(\mathbf{z}, \langle \mathbf{x}, \mathbf{z} \rangle + \tilde{x})$ on its own. The disadvantage of this approach is that in order to be useful, the modulus q and inverse error rate $1/\alpha$ typically must be super-polynomially large in n, which corresponds to assuming the worst-case hardness of various lattice problems for super-polynomial approximation factors and running times.

We also point out that in the above setting, if the ratio $\beta q/r$ is polynomial in n and a sufficiently large $h = \mathrm{poly}(n)$ number of hints are given out, then extended-LWE is *easy* to solve. To see this, view the h hints as $(\mathbf{Z} \in \mathbb{Z}^{m \times h}, \mathbf{y}^t := \mathbf{x}^t\mathbf{Z} + \tilde{\mathbf{x}}^t)$. With overwhelming probability, the singular values of $\mathbf{Z}$ will all be $r \cdot \Omega(\sqrt{h} - C\sqrt{m})$ for some universal constant $C > 0$ (see [Ver11, Theorem 5.39]). Thus, for sufficiently large $h = \mathrm{poly}(n)$, with overwhelming probability the singular values of the right-inverse $\mathbf{Z}^+ \in \mathbb{R}^{h \times m}$ of $\mathbf{Z}$ will all be small enough so that $\lfloor \tilde{\mathbf{x}}^t \cdot \mathbf{Z}^+ \rceil = \mathbf{0}$. As a result, we can compute $\lfloor \mathbf{y}^t\mathbf{Z}^+ \rceil = \mathbf{x}^t$, which trivially allows for solving the extended-LWE problem.

3.2 Reduction from LWE

Here we give a tight reduction from standard LWE to extended-LWE, which holds for the same parameters $n, q, \chi, m \geq n + \omega(\log n)$ in the two problems, and in which *no noise* is added to the hint $\langle \mathbf{z}, \mathbf{x} \rangle$ (i.e., $\beta = 0$). Our reduction imposes one requirement on the parameters: for $\mathbf{x} \leftarrow \chi^m$ and $\mathbf{z} \leftarrow \tau$, we need it to be the case that $|\langle \mathbf{x}, \mathbf{z} \rangle| < p$ with overwhelming probability, where p is the smallest prime divisor of the modulus q. For example, if $\chi = D_{\mathbb{Z},\alpha q}$ and $\tau = D_{\mathbb{Z},r}^m$, by standard tail inequalities it suffices to have $\alpha q \cdot r\sqrt{m+n} \cdot \omega(\sqrt{\log n}) < p$. In other words, the LWE inverse error rate is $1/\alpha > (q/p) \cdot r\sqrt{m+n}$, which is only polynomial in n when q, r, m are.

Theorem 3.1. *There exists a probabilistic polynomial-time oracle machine (a simulator) $\mathcal{S}$ such that for any adversary $\mathcal{A}$,*

$$\mathbf{Adv}_{\mathsf{LWE}}(\mathcal{S}^{\mathcal{A}}) \geq \tfrac{1}{2p-1} \cdot \mathbf{Adv}_{\mathsf{ELWE}}(\mathcal{A}) - \operatorname{negl}(n),$$

where the parameters of the LWE *and extended-*LWE *problems satisfy the condition specified above.*

Proof. For the proof it is convenient to use the equivalent "knapsack" form of LWE, which is: given $\mathbf{H} \leftarrow \mathbb{Z}_q^{(m-n)\times m}$ and $\mathbf{c} \in \mathbb{Z}_q^{m-n}$, where $\mathbf{c}$ is either $\mathbf{c} = \mathbf{Hx}$ for $\mathbf{x} \leftarrow \chi^m$, or is uniformly random and independent of $\mathbf{H}$, determine (with non-$\operatorname{negl}(n)$ advantage) which is the case. The extended form of the problem also reveals a hint $(\mathbf{z}, \langle \mathbf{x}, \mathbf{z} \rangle + \tilde{x})$, analogously to extended-LWE. The equivalence between LWE and its knapsack form for $m \geq n + \omega(\log n)$, which also applies to their extended versions, has been noticed in several works; a proof appears in [MM11, Lemmas 4.8 and 4.9].

We construct the reduction $\mathcal{S}$ as follows. It receives an LWE instance (in knapsack form) $\mathbf{H} \in \mathbb{Z}_q^{(m-n)\times m}, \mathbf{c} \in \mathbb{Z}_q^{m-n}$. It samples $\mathbf{z} \leftarrow \tau$, $\mathbf{x}' \leftarrow \chi^m$, and $\mathbf{v} \leftarrow \mathbb{Z}_q^{m-n}$, then lets

$$\mathbf{H}' := \mathbf{H} - \mathbf{v}\mathbf{z}^t \in \mathbb{Z}_q^{(m-n)\times m}, \quad \mathbf{c}' = \mathbf{c} - \mathbf{v} \cdot \langle \mathbf{z}, \mathbf{x}' \rangle \in \mathbb{Z}_q^{m-n}.$$

It sends $(\mathbf{H}', \mathbf{b}', \mathbf{z}, \langle \mathbf{x}', \mathbf{z} \rangle)$ to $\mathcal{A}$ (an adversary for extended-LWE in knapsack form), and outputs what $\mathcal{A}$ outputs.

We now analyze the behavior of $\mathcal{S}$. First consider the case where $\mathbf{H}, \mathbf{c}$ are uniform and independent. Then it is clear that $\mathbf{H}', \mathbf{c}'$ are as well, and both $\mathbf{x}'$ and $\mathbf{z}$ are also chosen exactly as in $\mathsf{ExptUnif}$, so $\mathcal{S}$ perfectly simulates $\mathsf{ExptUnif}$ to $\mathcal{A}$.

Now, consider the case where $\mathbf{H}, \mathbf{c}$ are drawn from the knapsack distribution, with $\mathbf{c} = \mathbf{Hx}$ for $\mathbf{x} \leftarrow \chi^m$. In this case, we have that $\mathbf{H}'$ is uniformly random (solely over the choice of $\mathbf{H}$), and

$$\mathbf{c}' = \mathbf{Hx} - \mathbf{v} \cdot \langle \mathbf{z}, \mathbf{x}' \rangle = \mathbf{H}'\mathbf{x} + \mathbf{v} \cdot \langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle.$$

So in the event that $\langle \mathbf{x}', \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle$, we have $\mathbf{c}' = \mathbf{H}'\mathbf{x}$ and so $\mathcal{S}$ perfectly simulates $\mathsf{ExptLWE}$ to $\mathcal{A}$. Whereas if $\langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle$ is a unit modulo q, then for any fixed choice of $\mathbf{H}'$, $\mathbf{z}$, $\mathbf{x}$, and $\mathbf{x}'$ (subject to this condition), we have that $\mathbf{c}'$ is uniformly random over the choice of $\mathbf{v}$ alone. Finally, since $\mathbf{x}$ and $\mathbf{x}'$ are identically distributed, it follows that $\mathcal{S}$ perfectly simulates $\mathsf{ExptUnif}$ to $\mathcal{A}$.

It remains to analyze the probabilities that $\langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle$ is zero or a unit (modulo q), respectively. First, by assumption $|\langle \mathbf{z}, \mathbf{x} - \mathbf{x}' \rangle| < p$ with overwhelming probability, so exactly one of the two cases holds; moreover, we have $\langle \mathbf{x}, \mathbf{z} \rangle = \langle \mathbf{x}', \mathbf{z} \rangle$ with probability at least $\frac{1}{2p-1} - \operatorname{negl}(n)$ because $\mathbf{x}$ and $\mathbf{x}'$ are independent. The theorem then follows from a routine calculation. □

Normal form. In our cryptosystems, we need to assume the hardness of extended-LWE in "normal form" (as in [MR09, ACPS09]), where the secret $\mathbf{s} \leftarrow \chi^n$ is drawn from the *error* distribution, the matrix $\mathbf{A}$ and vector $\mathbf{b}^t$ have $m - n$ columns, and the hint is of the form $\mathbf{z} \leftarrow \tau$, $b' = \langle (\mathbf{s}, \mathbf{x}), \mathbf{z} \rangle \in \mathbb{Z}$. Suppose m is sufficiently large so that a uniformly random matrix from $\mathbb{Z}_q^{n\times m}$ contains an invertible n-by-n submatrix with overwhelming probability. Then the reduction from [MR09, ACPS09] applies to extended-LWE in this form, with the slight modification that LWE samples in the first phase are never "thrown away" but are instead recycled to the second phase.

4 KDM-CPA Secure Public-Key Scheme

Here we present a "dual"-style LWE cryptosystem that is KDM-CPA secure for affine functions of the secret keys. In fact, by setting the parameters appropriately, the construction and security proof also encompass (a slight variant of) the cryptosystem from [LP11], which has somewhat smaller keys and ciphertexts than "primal" or "dual" systems. In Section 6 we build a KDM-CPA secure IBE around this system.

4.1 Construction

The cryptosystem involves a few parameters: a modulus $q = p^2$ for a prime p where the message space is $\mathbb{Z}_p$; integer dimensions n, m relating to the underlying LWE problems; and a Gaussian parameter r for key generation and encryption. To make embedding this scheme into our IBE more natural, Gen includes an additional parameter d, which will be used to specify the size of identity cliques in the IBE scheme, and outputs public keys $\mathbf{A}$ that are md columns wide. In the public-key scheme alone, the value d is unrelated to the number of public keys that the adversary can obtain in an attack (which is unbounded), and we would just fix $d = 1$.

- $\mathsf{Gen}(1^n, d)$: choose $\mathbf{A} \in \mathbb{Z}_q^{n \times md}$, $\mathbf{z}_0 \leftarrow D_{\mathbb{Z},r}^n$, $\mathbf{z}_1 \leftarrow D_{\mathbb{Z},r}^{md}$, and let $\mathbf{y} = \mathbf{z}_0 - \mathbf{A}\mathbf{z}_1 = [\mathbf{I}_n \mid -\mathbf{A}]\mathbf{z} \in \mathbb{Z}_q^n$ where $\mathbf{z} = (\mathbf{z}_0, \mathbf{z}_1) \in \mathbb{Z}^{n+md}$. The public key is $(\mathbf{A}, \mathbf{y})$ and the secret key is $\mathbf{z}_1$.

  (Notice that, unlike the dual-style encryption of [GPV08], but like the scheme of [LP11], the public key component $\mathbf{y}$ is a *perturbed* value of $-\mathbf{A}\mathbf{z}_1$. This will be important in the proof of KDM security.)
- $\mathsf{Enc}(\mathbf{A}, \mathbf{y}, \mu)$: to encrypt a message $\mu \in \mathbb{Z}_p$, choose $\mathbf{x}_0 \leftarrow D_{\mathbb{Z},r}^n$, $\mathbf{x}_1 \leftarrow D_{\mathbb{Z},r}^{md}$ and $x' \leftarrow D_{\mathbb{Z},r}$. Output the ciphertext $\mathbf{c}^t = \mathbf{x}_0^t[\mathbf{A} \mid \mathbf{y}] + [\mathbf{x}_1^t \mid x'] + [\mathbf{0} \mid p \cdot \mu] \in \mathbb{Z}_q^{1 \times (md+1)}$.
- $\mathsf{Dec}(\mathbf{z}_1, \mathbf{c})$: Compute $\mu' = \mathbf{c}^t \left[\begin{smallmatrix} \mathbf{z}_1 \\ 1 \end{smallmatrix}\right] \in \mathbb{Z}_q$. Output the $\mu \in \{0, \ldots, p-1\} = \mathbb{Z}_p$ such that μ' is closest to $(p\mu) \bmod q$.

4.2 Parameters and Correctness

For the public-key system alone, it suffices to take $m \geq n$ by our use of the extended-LWE assumption and its proof of hardness as in Section 3. When embedding the system into an IBE scheme, however, we will use $m = \Theta(n \log q)$ because we need the public parameters to be statistically close to uniform over the choice of the master secret key. The error parameter r must be small enough (relative to q/p) so that decryption is correct with overwhelming probability, but large enough to satisfy the reductions to LWE from worst-case lattice problems [Reg05, Pei09]; for the latter purpose, $r \geq 2\sqrt{n}$ suffices. (Note that even if part of the security proof relies on LWE in dimension $> n$, this problem is no easier than LWE in dimension n, and so we can still securely use $r = 2\sqrt{n}$ with the larger dimension.)

Here we give some example bounds. Let $r = 2\sqrt{n}$, let

$$p = r^2\sqrt{n + md} \cdot \omega(\sqrt{\log n}) = n\sqrt{n + md} \cdot \omega(\sqrt{\log n}),$$

and let $q = p^2$. Then decryption is correct except with probability $\mathrm{negl}(n)$: let $(\mathbf{A}, \mathbf{y}, \mathbf{z}) \leftarrow \mathsf{Gen}(1^n, d)$. For a ciphertext $\mathbf{c} \leftarrow \mathsf{Enc}(\mathbf{A}, \mathbf{y}, \mu)$, we have

$$\mathbf{c}^t \left[\begin{smallmatrix} \mathbf{z}_1 \\ 1 \end{smallmatrix}\right] = \mathbf{x}_0^t\mathbf{A}\mathbf{z}_1 + \langle \mathbf{x}_1, \mathbf{z}_1 \rangle + \langle \mathbf{x}_0, \mathbf{y} \rangle + x' + p \cdot \mu = \langle \mathbf{x}_0, \mathbf{z}_0 \rangle + \langle \mathbf{x}_1, \mathbf{z}_1 \rangle + x' + p \cdot \mu \bmod q,$$

so decryption is correct whenever $|\langle \mathbf{x}_0, \mathbf{z}_0 \rangle + \langle \mathbf{x}_1, \mathbf{z}_1 \rangle + x'| < p/2$. By known tail bounds on discrete Gaussians, this bound holds except with probability $\mathrm{negl}(n)$ (over the choice of all the random variables), as required.

4.3 Proof of Security

Theorem 4.1. *The above cryptosystem is KDM-CPA secure with respect to the set of affine functions over $\mathbb{Z}_p$, under the extended-LWE assumption for parameters described above.*

Proof. We proceed by a series of indistinguishable games. We begin with the real KDM-CPA attack game as defined in Section 2.5, where $\beta \in \{0,1\}$ is arbitrary. We then eventually transition to a game which proceeds in a manner independent of the value of β, thus proving computational indistinguishability between the attack games for $\beta = 0$ and $\beta = 1$. Since we are describing the games for an arbitrary value of β, we let $f = f_\beta$ denote the function of the secret key that is encrypted in each ciphertext query.

Game 0. This is the actual attack game. We do everything normally, generating public keys $(\mathbf{A}_i, \mathbf{y}_i)$ with secret keys $\mathbf{z}_{i,1}$ for any number of users, and encrypt affine functions $f_{\mathbf{v},w}(\mathbf{z}_{j,1}) = \langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w$ of the secret key $\mathbf{z}_{j,1}$ under the public key for user i as

$$\mathbf{c}^t = \mathbf{x}_0^t[\mathbf{A}_i \mid \mathbf{y}_i] + [\mathbf{x}^t \mid x'] + [\mathbf{0} \mid p \cdot (\langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w)].$$

Game 1. In this game we rewrite the way we respond to each key-dependent message query $f_{\mathbf{v},w}$, so that the response has exactly the same distribution, but it is constructed to syntactically match the $\mathsf{ExptLWE}$ experiment in the (amortized, normal form) extended-LWE problem. Looking ahead, in Game 2 we will switch the form of the ciphertexts to match the $\mathsf{ExptUnif}$ experiment.

For each user i, we proceed as in $\mathsf{ExptLWE}$ with parameters error distribution $\chi := D_{\mathbb{Z},r}$, extra noise $\beta := 0$, and the same r as in the cryptosystem. That is, we choose $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n \times md}$ and $\mathbf{z}_i \leftarrow D_{\mathbb{Z},r}^{n+md}$, parsing it as $\mathbf{z}_i = (\mathbf{z}_{i,0}, \mathbf{z}_{i,1}) \in \mathbb{Z}^n \times \mathbb{Z}^{md}$. We let $\mathbf{y}_i = [\mathbf{I}_n \mid -\mathbf{A}_i]\mathbf{z}_i = \mathbf{z}_{i,0} - \mathbf{A}_i\mathbf{z}_{i,1}$, giving user i's public key $(\mathbf{A}_i, \mathbf{y}_i)$ to the adversary. (As expected, $\mathbf{z}_{i,1}$ is the secret key for user i.)

For handling later ciphertext queries, we also (lazily) generate an unbounded number of LWE vectors $\mathbf{b}$, with hints b', of the form

$$\mathbf{b}^t := \mathbf{x}_0^t\mathbf{A}_i + \mathbf{x}_1^t \in \mathbb{Z}_q^{1 \times md}, \quad b' := \langle \mathbf{x}, \mathbf{z}_i \rangle \in \mathbb{Z},$$

where $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \leftarrow D_{\mathbb{Z},r}^{n+md}$ is freshly chosen for each $\mathbf{b}$, and are amortized over the same fixed $\mathbf{A}_i$, $\mathbf{z}_i$ chosen above.

To answer a query to encrypt $f_{\mathbf{v},w}(\mathbf{z}_{j,1})$ under user i's public key, we generate a fresh $\mathbf{b}$ with hint b' as above, choose $x' \leftarrow D_{\mathbb{Z},r}$ as usual, and return to the adversary the ciphertext

$$\mathbf{c}^t = [\mathbf{b}^t \mid -\langle \mathbf{b}, \mathbf{z}_{i,1} \rangle + b' + x' + p \cdot (\langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w)].$$

We now show that the responses to the KDM queries in this game are distributed exactly the same as in the previous game. To see this, simply observe that each ciphertext

$$\begin{aligned}
\mathbf{c}^t &= [\mathbf{b}^t \mid -\langle \mathbf{b}, \mathbf{z}_{i,1} \rangle + b' + x' + p \cdot (\langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w)] \\
&= [\mathbf{x}_0^t\mathbf{A}_i + \mathbf{x}_1^t \mid -(\mathbf{x}_0^t\mathbf{A}_i + \mathbf{x}_1^t)\mathbf{z}_{i,1} + \langle \mathbf{x}, \mathbf{z}_i \rangle + x' + p \cdot (\langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w)] \\
&= [\mathbf{x}_0^t\mathbf{A}_i + \mathbf{x}_1^t \mid \langle \mathbf{x}_0, \mathbf{y}_i \rangle + x' + p \cdot (\langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w)],
\end{aligned}$$

which has the same distribution as in Game 0.

Game 2. This game is exactly like Game 1, except that instead of using $\mathsf{ExptLWE}$, we instead use $\mathsf{ExptUnif}$, i.e., we choose the vectors $\mathbf{b}$ uniformly at random (but still choose the vectors $\mathbf{x}$ for generating the hints $b' = \langle \mathbf{x}, \mathbf{z}_i \rangle$). Responses to key-dependent message queries are still generated as

$$\mathbf{c}^t = [\mathbf{b}^t \mid -\langle \mathbf{b}, \mathbf{z}_{i,1} \rangle + x' + b' + p \cdot (\langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w)].$$

Under the extended-LWE assumption, and hence under the standard LWE assumption with appropriate parameters (by Theorem 3.1), this game is computationally indistinguishable from Game 1.

Game 3. In this game, we use the LWE distribution transformation from [ACPS09] (our Lemma 2.4) to answer key-dependent queries without needing to use the secret keys $\mathbf{z}_{i,1}$ explicitly. We begin with oracle access to $A_{\mathbf{t},\chi}$ (where $\mathbf{t} \leftarrow \mathbb{Z}_q^{md}$, $\chi = D_{\mathbb{Z},r}$). We apply the transformation for each i, yielding oracle access to distributions $A_{\mathbf{z}_{i,1},\chi}$, where each $\mathbf{z}_{i,1}$ is distributed according to χ^{md}. For each user i, we sample $A_{\mathbf{z}_{i,1},\chi}$ a total of n times to get $(-\mathbf{A}_i \in \mathbb{Z}_q^{n \times md}, \mathbf{y}_i = \mathbf{z}_{i,0} - \mathbf{A}_i \mathbf{z}_{i,1})$, and set the public key for user i to be $(\mathbf{A}_i, \mathbf{y}_i)$. (We write $-\mathbf{A}_i$ instead of $\mathbf{A}_i$ so that $\mathbf{y}_i$ has the same form with respect to $\mathbf{A}_i$ as it does in the actual encryption scheme.)

As in [ACPS09], we use the auxiliary information output by the LWE transformation to construct for all users i, j a linear relation between the (unknown) secret keys $\mathbf{z}_{i,1}$, $\mathbf{z}_{j,1}$, i.e., an invertible matrix $\mathbf{T}_{i,j} \in \mathbb{Z}_q^{md \times md}$ and vector $\mathbf{w}_{i,j} \in \mathbb{Z}_q^{md}$ such that

$$\mathbf{z}_{j,1} = \mathbf{T}_{i,j}^t \cdot \mathbf{z}_{i,1} + \mathbf{w}_{i,j} \bmod q. \tag{4.1}$$

To respond to a query for an encryption of $f_{\mathbf{v},w}(\mathbf{z}_{j,1}) = \langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w$ under the public key for user i, we draw a fresh $(-\mathbf{b}_0, b_1) \leftarrow A_{\mathbf{z}_{i,1},\chi}$. Next, we choose a fresh $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \leftarrow D_{\mathbb{Z},r}^{n+md}$. We output the ciphertext

$$\mathbf{c}^t = [\mathbf{b}_0^t + (\mathbf{x}_0^t \mathbf{A}_i + \mathbf{x}_1^t) + p \cdot (\mathbf{T}_{i,j}\mathbf{v})^t \mid b_1 + \langle \mathbf{x}_0, \mathbf{y}_i \rangle + p \cdot (\langle \mathbf{v}, \mathbf{w}_{i,j} \rangle + w)].$$

We claim that these ciphertexts are distributed identically to those in Game 2. To show this, let $\mathbf{b}^t = \mathbf{b}_0^t + (\mathbf{x}_0^t \mathbf{A}_i + \mathbf{x}^t) + p \cdot (\mathbf{T}_{i,j}\mathbf{v})^t$. Since $\mathbf{b}_0$ is uniform (by the definition of $A_{\mathbf{z}_i,\chi}$), $\mathbf{b}$ in this game is uniform over the choice of $\mathbf{b}_0$ alone, just as in Game 2. Next, since $b_1 = -\langle \mathbf{b}_0, \mathbf{z}_i \rangle + x'$ for $x' \leftarrow D_{\mathbb{Z},r}$, we can rewrite the ciphertext constructed in this game as

$$\begin{aligned}
\mathbf{c}^t &= [\mathbf{b}^t \mid \langle -\mathbf{b} + (\mathbf{A}_i^t \mathbf{x}_0 + \mathbf{x}_1) + p \cdot \mathbf{T}_{i,j}\mathbf{v}, \mathbf{z}_{i,1} \rangle + x' + \langle \mathbf{x}_0, \mathbf{y}_i \rangle + p \cdot (\langle \mathbf{v}, \mathbf{w}_{i,j} \rangle + w)] \\
&= [\mathbf{b}^t \mid -\langle \mathbf{b}, \mathbf{z}_{i,1} \rangle + x' + \langle \mathbf{x}_0, \mathbf{y}_i + \mathbf{A}_i \mathbf{z}_{i,1} \rangle + \langle \mathbf{x}_1, \mathbf{z}_{i,1} \rangle + p \cdot \langle \mathbf{v}, \mathbf{z}_{j,1} - \mathbf{w}_{i,j} \rangle + p \cdot (\langle \mathbf{v}, \mathbf{w}_{i,j} \rangle + w)] \\
&= [\mathbf{b}^t \mid -\langle \mathbf{b}, \mathbf{z}_{i,1} \rangle + x' + \langle \mathbf{x}_0, \mathbf{z}_{i,0} \rangle + \langle \mathbf{x}_1, \mathbf{z}_{i,1} \rangle + p \cdot (\langle \mathbf{v}, \mathbf{z}_{j,1} \rangle + w)],
\end{aligned}$$

which is distributed exactly as in Game 2.

Game 4. This game is exactly like Game 3, except instead of beginning with oracle access to $A_{\mathbf{t},\chi}$ for $\chi = D_{\mathbb{Z},r}$, we begin with oracle access to $U(\mathbb{Z}_q^{md} \times \mathbb{Z}_q)$. Computational indistinguishability from Game 3 follows directly from the hardness of decision-LWE.

In this game, the public keys $(\mathbf{A}_i, \mathbf{y}_i)$ are now all uniformly random and independent. Moreover, since every sample $(\mathbf{b}_0, b_1)$ drawn in the game is also truly uniform and independent of everything else in the ciphertexts, all of the responses to key-dependent message queries are just uniformly random and independent vectors. This makes the game independent of β, and concludes the proof. □

5 All-But-d Trapdoor Functions

Here we develop a technique for constructing "all-but-d" (tag-based) trapdoor functions, which, informally, are trapdoor functions for which the trapdoor enables efficient inversion for all but (up to) d tags, which are specified at the time of key generation. This is the main tool we use for embedding our KDM-CPA-secure public-key cryptosystem into an identity-based encryption scheme.

Our construction is a generalization (to higher-degree polynomials) of the main technique from [ABB10]. For simplicity and somewhat better efficiency, we follow the construction of [MP12], specifically the use of a fixed, public "gadget" matrix $\mathbf{G}$ as described in Section 2.2.

5.1 Algebraic Background

Let $n \geq 1$, $q \geq 2$, and $d = \text{poly}(n)$ be integers. Let $\mathcal{R}$ denote any commutative ring (with efficiently computable operations, including inversion of multiplicative units) such that the additive group $\mathbb{G} = \mathbb{Z}_q^n$ is an $\mathcal{R}$-module, and such that there are at least $d+1$ known elements $U = \{u_0 = 0, u_1, u_2, \ldots\} \subseteq \mathcal{R}$ where $u_i - u_j$ is invertible in $\mathcal{R}$ (i.e., a unit) for every $i \neq j$. (These are the only abstract properties of the ring we will need for our constructions. In the next paragraph we recall how to construct such a ring for any n and q, where U has size at least 2^n.) In particular, we have an (efficiently computable) scalar multiplication operation $\mathcal{R} \times \mathbb{G} \to \mathbb{G}$. Note that multiplication by $u \in \mathcal{R}$ is an invertible linear transformation on $\mathbb{G}$ exactly when u is invertible (i.e., a unit). We extend scalar multiplication in the natural way to vectors and matrices, i.e., $R^{a \times b} \times \mathbb{G}^{b \times c} \to \mathbb{G}^{a \times c}$. To avoid confusion with vectors and matrices over $\mathbb{Z}_q$, we use $\vec{u}$ notation for vectors over $\mathcal{R}$, and V notation for matrices over $\mathcal{R}$.

To construct a suitable ring, we use ideas from the literature on secret sharing over groups and modules, e.g., [DF94, Feh98]. We use an extension ring $\mathcal{R} = \mathbb{Z}_q[x]/(F(x))$ for any monic, degree-n, irreducible $F(x) = F_0 + F_1 x + \cdots + F_{n-1}x^{n-1} + x^n \in \mathbb{Z}_q[x]$. Scalar multiplication $\mathcal{R} \times \mathbb{G} \to \mathbb{G}$ is defined by identifying each $\mathbf{a} = (a_0, \ldots, a_{n-1})^t \in \mathbb{G}$ with the polynomial $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathcal{R}$, multiplying in $\mathcal{R}$, then mapping back to $\mathbb{G}$. In other words, scalar multiplication is defined by the linear transformation $x \cdot (a_0, \ldots, a_{n-1})^t = (0, a_0, \ldots, a_{n-2})^t - a_{n-1}(F_0, F_1, \ldots, F_{n-1})^t$. It is easy to check that with this scalar product, $\mathbb{G}$ is an $\mathcal{R}$-module. In addition, by the Chinese remainder theorem, $r \in \mathcal{R}$ is a unit if and only if it is nonzero (as a polynomial residue) modulo every prime integer divisor p of q. (This is because $\mathbb{Z}_p[x]/(F(x))$ is a field by construction.) Letting p be the smallest such divisor of q, we can define the universe $U = \{u_0 = 0, u_1, u_2, \ldots\} \subseteq \mathcal{R}$ to consist of all the polynomial residues having coefficients in $\{0, \ldots, p-1\}$. Then $|U| = p^n \geq 2^n$ and $u_i - u_j$ is a unit for all $i \neq j$, as desired.

5.2 Basic Construction

As in [MP12], we fix a universal public "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times w}$ for which there is an efficient Gaussian preimage sampling algorithm for parameter $s \geq \omega(\sqrt{\log n})$, i.e., an algorithm that given any $\mathbf{u} \in \mathbb{Z}_q^n$ outputs a sample from $D_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{G}),s}$. E.g., we can let $\mathbf{G} = \mathbf{I}_n \otimes (1, 2, 4, \ldots, 2^{k-1}) \in \mathbb{Z}_q^{n \times nk}$ for $k = \lceil \lg q \rceil$.

As input, the trapdoor generator takes:

- an integer $d \geq 1$ and a monic degree-d polynomial $f(z) = c_0 + c_1 z + \cdots + z^d \in \mathcal{R}[z]$,
- a (usually uniformly random) matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{(nd) \times \bar{m}}$ for some $\bar{m} \geq 1$, which is made up of stacked submatrices $\bar{\mathbf{A}}_i \in \mathbb{Z}_q^{n \times \bar{m}}$ for $i = 0, \ldots, d-1$.
- a "short" secret $\mathbf{R} \in \mathbb{Z}^{\bar{m} \times w}$ chosen at random from an appropriate distribution (typically, a discrete Gaussian) to serve as a trapdoor.

As output it produces a matrix $\mathbf{A} \in \mathbb{Z}_q^{(nd)\times(\bar{m}+w)}$ (which is statistically close to uniform, when the parameters and input $\bar{\mathbf{A}}$ are appropriately chosen). In addition, for each tag $u \in U$ there is an efficiently computable (from $\mathbf{A}$) matrix $\mathbf{A}_u \in \mathbb{Z}_q^{n\times(\bar{m}+w)}$ for which $\mathbf{R}$ *may* be a trapdoor, depending on the value of $f(u) \in \mathcal{R}$.

We write the coefficients of $f(z)$ as a column vector $\vec{c} = (c_0, c_1, \ldots, c_{d-1})^t \in \mathcal{R}^d$, and define

$$\mathbf{A}'_f := \begin{bmatrix} \bar{\mathbf{A}} & \vec{c} \otimes \mathbf{G} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}}_0 & c_0 \cdot \mathbf{G} \\ \vdots & \vdots \\ \bar{\mathbf{A}}_{d-1} & c_{d-1} \cdot \mathbf{G} \end{bmatrix} \in \mathbb{Z}_q^{(nd)\times(\bar{m}+w)}.$$

To hide the polynomial f, we output the public key

$$\mathbf{A} := \mathbf{A}'_f \cdot \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{A}} & (\vec{c} \otimes \mathbf{G}) - \bar{\mathbf{A}}\mathbf{R} \end{bmatrix}.$$

Note that as long as the distribution of $[\bar{\mathbf{A}} \mid -\bar{\mathbf{A}}\mathbf{R}]$ is statistically close to uniform, then so is $\mathbf{A}$ for any f.

The tag space for the trapdoor function is the set $U \subset \mathcal{R}$. For any tag $u \in U$, define the row vector $\vec{u}^t := (u^0, u^1, \cdots, u^{d-1}) \in \mathcal{R}^d$ (where $0^0 = 1$) and the derived matrix for tag u to be

$$\mathbf{A}_u := \vec{u}^t \cdot \mathbf{A} + \begin{bmatrix} \mathbf{0} & u^d \cdot \mathbf{G} \end{bmatrix} = \begin{bmatrix} \vec{u}^t \cdot \bar{\mathbf{A}} & f(u) \cdot \mathbf{G} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix}.$$

By the condition in Lemma 2.3, $\mathbf{R}$ is a (strong) trapdoor for $\mathbf{A}_u$ exactly when $f(u) \in \mathcal{R}$ is a unit, because $\mathbf{A}_u \cdot \left[\begin{smallmatrix} \mathbf{R} \\ \mathbf{I} \end{smallmatrix}\right] = f(u) \cdot \mathbf{G}$ and $f(u)$ represents an invertible linear transformation when it is a unit.

5.3 Puncturing

In our cryptosystems and security proofs we will need to generate (using the above procedure) an all-but-d trapdoor function that is "punctured" at up to d tags. More precisely, we are given as input:

- a set of distinct tags $P = \{u_1, \ldots, u_\ell\} \subseteq U$ for some $\ell \leq d$,
- uniformly random matrices $\mathbf{A}_i^* \in \mathbb{Z}_q^{n\times\bar{m}}$ for $i \in [\ell]$ (which often come from an SIS or LWE challenge),
- a "short" secret $\mathbf{R} \in \mathbb{Z}^{\bar{m}\times w}$ chosen at random from an appropriate distribution (typically, a discrete Gaussian) to serve as a trapdoor,
- optionally, some uniformly random auxiliary matrices $\mathbf{Y}_i^* \in \mathbb{Z}_q^{n\times k}$ for $i \in [\ell]$ and some $k \geq 0$.

As output we produce a public key $\mathbf{A} \in \mathbb{Z}_q^{(nd)\times\bar{m}}$ and auxiliary matrix $\mathbf{Y} \in \mathbb{Z}_q^{(nd)\times k}$ so that:

1. Each $\mathbf{A}_{u_i}$ matches the challenge matrix $\mathbf{A}_i^*$, and $\mathbf{R}$ is only a "weak" trapdoor for $\mathbf{A}_{u_i}$. More precisely,

$$\mathbf{A}_{u_i} = \begin{bmatrix} \mathbf{A}_i^* & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix}.$$

2. $\mathbf{R}$ is a (strong) trapdoor for $\mathbf{A}_u$ for any *nonzero* $u \in U \setminus P$, i.e., $f(u)$ is a unit.
3. The auxiliary matrix $\mathbf{Y}_{u_i} := \vec{u_i}^t \cdot \mathbf{Y}$ equals the auxiliary input $\mathbf{Y}_i^*$ for each $u_i \in P$.

We satisfy these criteria by invoking the above trapdoor generator with the following inputs f and $\bar{\mathbf{A}}$:

1. We define the monic degree-d polynomial

$$f(z) = z^{d-\ell} \cdot \prod_{i \in [\ell]} (z - u_i) \in \mathcal{R}[z]$$

   and expand to compute its coefficients $c_i \in \mathcal{R}$. Note that $f(u_i) = 0$ for every $u_i \in P$, and $f(u)$ is a unit for any nonzero $u \in U \setminus P$ because $0 \in U$ and $u_i - u_j$ is a unit for every distinct $u_i, u_j \in U$.

2. We define $\bar{\mathbf{A}}$ using interpolation: let $\mathbf{A}^* \in \mathbb{Z}_q^{(n\ell) \times \bar{m}}$ denote the stack of challenge matrices $\mathbf{A}_i^*$, and let $V \in \mathcal{R}^{\ell \times d}$ be the Vandermonde matrix whose rows are the vectors $\vec{u_i}^t$ defined above. We then let $\bar{\mathbf{A}} \in \mathbb{Z}_q^{(nd) \times \bar{m}}$ be a uniformly random solution to $V \cdot \bar{\mathbf{A}} = \mathbf{A}^*$.

   Such a solution exists, and is efficiently computable and uniformly random (over the uniformly random choice of $\mathbf{A}^*$ and the random solution chosen). To see this, extend V to an invertible $d \times d$ Vandermonde matrix over $\mathcal{R}$ having unit determinant $\prod_{i<j}(u_j - u_i) \in \mathcal{R}^*$, by adding $d - \ell$ additional rows $\vec{u_j}^t$ for arbitrary distinct $u_j \in U \setminus P$. Likewise, extend $\mathbf{A}^*$ to have dimension $(nd) \times \bar{m}$ by adding uniformly random rows. Then for any fixed choice of the (extended) matrix V, the (extended) matrix $\mathbf{A}^*$ and solution $\bar{\mathbf{A}}$ are in bijective correspondence, and so the latter is uniformly random because the former is.

3. We also define the auxiliary matrix $\mathbf{Y}$ similarly using interpolation, as a uniformly random solution to $V \cdot \mathbf{Y} = \mathbf{Y}^*$.

6 Circular-Secure IBE

Our IBE scheme is a generalization of the efficient IBE scheme of Agrawal *et al.* [ABB10]. Other than some minor changes in the parameters, the main difference is the use of the all-but-d trapdoor construction, which allows us to "puncture" the master public key at up to d identities in the security proof. The scheme has parameters modulus q, message space $\mathbb{Z}_p$ for some $p < q$, dimension m, and Gaussian parameters r and γ. We give example instantiations after describing the scheme.

6.1 Construction

The identity space for the scheme is $U \setminus \{0\} \subset \mathcal{R}$, where U, $\mathcal{R}$ are constructed as in Section 5.

- $\mathsf{Setup}(1^n, d)$: On input security parameter 1^n and secret key clique size d:

  1. Sample $\mathbf{R} \leftarrow D^{md \times w}_{\mathbb{Z}, \omega(\sqrt{\log n})}$, and for $i = 0, \ldots, d-1$, choose uniformly random $\mathbf{A}_i \leftarrow \mathbb{Z}_q^{n \times md}$, $\mathbf{y}_i \leftarrow \mathbb{Z}_q^n$ and let $\tilde{\mathbf{A}}_i = -\mathbf{A}_i \mathbf{R} \in \mathbb{Z}_q^{n \times w}$. (Note that this is simply calling the all-but-d trapdoor construction from Section 5 with an empty set of punctured tags.) Let

$$\mathbf{A} := \begin{bmatrix} \mathbf{A}_0 \\ \vdots \\ \mathbf{A}_{d-1} \end{bmatrix}, \quad \tilde{\mathbf{A}} := \begin{bmatrix} \tilde{\mathbf{A}}_0 \\ \vdots \\ \tilde{\mathbf{A}}_{d-1} \end{bmatrix} = -\mathbf{A}\mathbf{R}, \quad \mathbf{y} := \begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_{d-1} \end{bmatrix}.$$

  2. The public key is $mpk = (\mathbf{A}, \tilde{\mathbf{A}}, \mathbf{y})$. The master secret key is $msk = (\mathbf{R})$.

- $\mathsf{Ext}(mpk, msk, u)$ On input mpk, msk and $u \in U \setminus \{0\} \subseteq \mathcal{R}$:

1. Let $\vec{u}^t := (u^0, u^1, \ldots, u^{d-1})$, $\bar{\mathbf{A}}_u = \vec{u}^t \cdot \mathbf{A}$, $\mathbf{y}_u = \vec{u}^t \cdot \mathbf{y}$ and $\mathbf{A}_u = [\bar{\mathbf{A}}_u \mid u^d\mathbf{G} - \bar{\mathbf{A}}_u\mathbf{R}]$, as in Section 5.
2. Sample $\mathbf{z}_0 \leftarrow D^n_{\mathbb{Z},r}$, $\mathbf{z}_1 \leftarrow D_{\Lambda^\perp_{\mathbf{z}_0 - \mathbf{y}_u}(\mathbf{A}_u),r}$ using the preimage sampling algorithm (Lemma 2.3), so that $\mathbf{y}_u = \mathbf{z}_0 - \mathbf{A}_u\mathbf{z}_1$ (as in the public-key cryptosystem from Section 4). Output $sk_u := \mathbf{z}_1$. Note that the above is possible because $u^d \in \mathcal{R}$ is a unit, and by our choice of r below, because $s_1(\mathbf{R}) = O(\sqrt{md} + \sqrt{w}) \cdot \omega(\sqrt{\log n}) = O(\sqrt{md}) \cdot \omega(\sqrt{\log n})$ with all but $\mathrm{negl}(n)$ probability by Lemma 2.1.

- $\mathsf{Enc}(mpk, u, \mu)$: On input master public key, identity $u \in U \setminus \{0\}$, and message $\mu \in \mathbb{Z}_p$ do:
  1. Let $\vec{u}^t := (u^0, u^1, \ldots, u^{d-1})$, $\mathbf{A}_u = [\vec{u}^t \cdot \mathbf{A} \mid u^d\mathbf{G} + \vec{u}^t \cdot \tilde{\mathbf{A}}] \in \mathbb{Z}_q^{n \times md+w}$, and $\mathbf{y}_u = \vec{u}^t \cdot \mathbf{y}$.
  2. Choose $\mathbf{s} \leftarrow D^n_{\mathbb{Z},r}$, $\mathbf{x}_0 \leftarrow D^{md}_{\mathbb{Z},r}$, $\mathbf{x}_1 \leftarrow D^w_{\mathbb{Z},\gamma}$, $x_2 \leftarrow D_{\mathbb{Z},r}$. Let $\mathbf{x}^t = [\mathbf{x}_0^t \mid \mathbf{x}_1^t]$.
  3. Output the ciphertext $\mathbf{c}^t = \mathbf{s}^t[\mathbf{A}_u \mid \mathbf{y}_u] + [\mathbf{x}^t \mid x_2] + [\mathbf{0} \mid p \cdot \mu]$.
- $\mathsf{Dec}(mpk, sk_u = \mathbf{z}_1, \mathbf{c})$: output the $\mu \in \mathbb{Z}_p$ such that $\mathbf{c}^t \left[\begin{smallmatrix}\mathbf{z}_1 \\ 1\end{smallmatrix}\right]$ is closest to $p \cdot \mu$ modulo q.

6.2 Parameters and Correctness

We need most of the parameters to match the parameters from the public-key encryption scheme, with the additional constraint that r must be large enough that we can run the preimage sampling algorithm (Lemma 2.3) in Ext. Thus, we choose a sufficiently large $m = \Theta(n \log q)$, $r = O(\sqrt{md}) \cdot \omega(\sqrt{\log n})^2$, $\gamma = n^{\omega(1)}$ slightly superpolynomial in n, $p = \gamma \cdot \mathrm{poly}(n)$ for a sufficiently large $\mathrm{poly}(n)$ term to ensure correctness, and $q = p^2$. We need $m = \Theta(n \log q)$ so that the stacked matrix $\mathbf{A} \in \mathbb{Z}_q^{nd \times md}$ is wide enough so that $(\mathbf{A}, \tilde{\mathbf{A}} = \mathbf{AR})$ is statistically close to uniform (by Lemma 2.1) over our choice of $\mathbf{A}$ and $\mathbf{R}$.

We now prove correctness. Let $\mathbf{c}^t \leftarrow \mathsf{Enc}(mpk, u, \mu)$ be a properly generated encryption of μ under identity u, and let $\mathbf{z}_1 \leftarrow \mathsf{Ext}(mpk, msk, u)$. Then we have

$$\mathbf{c}^t \left[\begin{smallmatrix}\mathbf{z}_1 \\ 1\end{smallmatrix}\right] = \mathbf{s}^t\mathbf{A}_u\mathbf{z}_1 + \mathbf{s}^t\mathbf{y}_u + \langle \mathbf{x}, \mathbf{z}_1 \rangle + x_2 + p \cdot \mu = \langle \mathbf{s}, \mathbf{z}_0 \rangle + \langle \mathbf{x}, \mathbf{z}_1 \rangle + x_2 + p \cdot \mu.$$

Thus, decryption will be correct whenever $|\langle \mathbf{s}, \mathbf{z}_0 \rangle + \langle \mathbf{x}, \mathbf{z}_1 \rangle + x_2| < p/2$. By Cauchy-Schwarz and Lemma 2.1, this bound holds except with probability negligible in n (over the choice of all the random variables), as required.

6.3 Proof of Security

Theorem 6.1. *For the above parameters, the above IBE scheme is selective identity KDM-CPA secure with respect to the set of affine functions over $\mathbb{Z}_p$, under the $\mathsf{LWE}_{q,\chi}$ assumption for $\chi = D_{\mathbb{Z},r}$, and the KDM-CPA security of the system from Section 4.*

Proof. Our proof of security proceeds as follows. Game 0 is the actual attack game. In Game 1, we use the all-but-d trapdoors construction from Section 5 to construct the master public key, "puncturing" it at the targeted identities. Finally, in Game 2, we play the KDM-CPA security game against a challenger running the public-key encryption scheme from Section 4 and use the outputs of the challenger to simulate Game 1. This requires some care because the IBE secret keys and ciphertexts have larger dimension by an additive term of w (the width of $\mathbf{G}$). To address this, we fill in the missing dimensions of the secret keys by choosing them ourselves, and use knowledge of the master secret key to fill in the missing dimensions of the ciphertexts (here is where we use the fact that noise with parameter γ "overwhelms" noise with parameter r). Selective identity KDM-CPA security then follows from the KDM-CPA security of the public-key encryption scheme.

Game 0. This is the actual security game from Section 2.5. For bit β, we respond to KDM queries $(f_0 = \langle \mathbf{v}_0, \mathbf{e}_{j_0} \rangle + w_0, f_1 = \langle \mathbf{v}_1, \mathbf{e}_{j_1} \rangle + w_1, i)$ by encrypting $f_\beta = \langle \mathbf{v}_\beta, \mathbf{e}_{k_\beta} \rangle + w_\beta$ under identity u_i. We respond with ciphertext $\mathbf{c}$, where (for $\vec{u}_i^t = (u_i^0, u_i^1, \ldots, u_i^{d-1})$)

$$\mathbf{c}^t = \mathbf{s}^t[\vec{u}_i^t \cdot \mathbf{A} \mid \vec{u}_i^t \cdot \tilde{\mathbf{A}} \mid \vec{u}_i^t \cdot \mathbf{y}] + [\mathbf{x}^t \mid x_2] + [\mathbf{0} \mid p \cdot (\langle \mathbf{v}_\beta, \mathbf{e}_{j_\beta} \rangle + w_\beta)].$$

Game 1. In this game, we use the all-but-d trapdoor construction from Section 5 to "puncture" the public key at each of the d challenge identities in a statistically indistinguishable manner.

We first choose d uniform random matrices $\mathbf{A}_i^* \in \mathbb{Z}_q^{n \times md}$ and master secret key $\mathbf{R} \leftarrow D_{\mathbb{Z},\omega(\sqrt{\log n})}^{md \times w}$. In order to successfully simulate the security game, we still need to know a secret key for each of the d challenge identities. So, for $i \in [d]$, we choose the secret key for identity u_i to be $\mathbf{e}_i \leftarrow D_{\mathbb{Z},r}^{md+w}$ and choose error $\mathbf{z}_0 \leftarrow D_{\mathbb{Z},r}^n$. We then set $\mathbf{y}_i^* = \mathbf{z}_0 - [\mathbf{A}_i^* \mid -\mathbf{A}_i^*\mathbf{R}]\mathbf{e}_i$.

Lemma 2.1 implies that this is statistically close to choosing the $\mathbf{y}_i^*$ uniformly at random and then sampling $\mathbf{e}_i \leftarrow D_{\Lambda^\perp_{\mathbf{z}_0 - \mathbf{y}_i^*}([\mathbf{A}_i^* | -\mathbf{A}_i^*\mathbf{R}]),r}$. Recalling that as a result of the all-but-d construction, we will have that $[\mathbf{A}_i^* \mid -\mathbf{A}_i^*\mathbf{R}] = [\vec{u}_i^t \cdot \mathbf{A} \mid u_i^d\mathbf{G} - \vec{u}_i^t \cdot \mathbf{AR}]$ and that $\mathbf{y}_i^* = \vec{u}_i^t \cdot \mathbf{y}$, we see that the master public key and the secret keys for the challenge identities have been generated in a manner statistically indistinguishable from how they were generated in Game 0.

So, we invoke the all-but-d trapdoor construction on $\mathbf{A}_i^*, \mathbf{y}_i^*, \mathbf{R}$ and identities u_i (that we received from the adversary), and receive back $\mathbf{A}_i, \tilde{\mathbf{A}}_i = -\mathbf{A}_i\mathbf{R} + c_i\mathbf{G}, \mathbf{y}$ for $i = 0, \ldots, d-1$ which we insert the master public key and give to the adversary.

Secret key extractions and responses to KDM queries proceed normally (using the now "punctured" public key), and so responses to KDM queries are now of the form

$$\mathbf{c}^t = \mathbf{s}^t[\mathbf{A}_i^* \mid -\mathbf{A}_i^*\mathbf{R} \mid \mathbf{y}_i^*] + [\mathbf{x}^t \mid x_2] + [\mathbf{0} \mid p \cdot (\langle \mathbf{v}_\beta, \mathbf{e}_{j_\beta} \rangle + w_\beta)]. \tag{6.1}$$

Game 2. In this game, we attack the KDM-CPA secure scheme described in Section 4 and use it to simulate Game 1. The secret keys in that construction only correspond to the "top parts" of the secret keys in the IBE scheme (denoted below as $\mathbf{e}_i^{(1)}$). The "bottom parts" will be denoted $\mathbf{e}_i^{(2)}$, with similar notation for the syndrome $\mathbf{y}_i$ and affine function vectors $\mathbf{v}_0, \mathbf{v}_1$. We will be choosing the "bottom parts" of the keys on our own, as described below.

After receiving the d challenge identities from the adversary, we begin playing the KDM-CPA security game (for the same values of m and d) against a challenger. The challenger sends us each user's public key as $(\mathbf{A}_i^*, (\mathbf{y}_i^*)^{(1)} = -\mathbf{A}_i^*\mathbf{e}_i^{(1)} + \tilde{\mathbf{e}}_i)$ for $i \in [d]$, where the $\mathbf{e}_i^{(1)}$ and $\tilde{\mathbf{e}}_i$ are secret. To construct the full syndrome $\mathbf{y}_i^*$ for each challenge identity u_i in the IBE scheme, we sample $\mathbf{e}_i^{(2)} \leftarrow D_{\mathbb{Z},r}^w$, compute $(\mathbf{y}_i^*)^{(2)} = \mathbf{A}_i^*\mathbf{R}\mathbf{e}_i^{(2)}$, and then set $\mathbf{y}_i^* = (\mathbf{y}_i^*)^{(1)} + (\mathbf{y}_i^*)^{(2)}$. Note that as in the previous game, the $\mathbf{A}_i^*$, $\mathbf{y}_i^*$ remain statistically close to uniform by Lemma 2.1.

We then sample $\mathbf{R} \leftarrow D_{\mathbb{Z},\omega(\sqrt{\log n})}^{md \times w}$ as usual, and use $(\mathbf{A}_i^*, \mathbf{R}, \mathbf{y}_i^*, u_i)$ as the input to our all-but-d trapdoor construction, receiving back $\mathbf{A}_i$, $\tilde{\mathbf{A}}_i = -\mathbf{A}_i\mathbf{R} + c_i\mathbf{G}$, and $\mathbf{y}_i$ for $i = 0, \ldots, d-1$ to insert into the master public key.

We continue to respond to secret key extraction queries as in Game 1, using our all-but-d trapdoor to extract secret keys for all non-challenge identities.

We now show how to respond to the adversary's key-dependent message queries with encryptions of f_β, where β parameterizes the KDM-CPA security game that we are playing (and is unknown to us because that scheme is KDM-CPA secure). To do so, we need to modify the functions f_0, f_1 given to us by the adversary before passing them along to the KDM-CPA challenger. For $i \in \{0,1\}$, our modifications will add to the constant part of each function (w_i) the inner product of the "bottom part" of each affine function vector ($\mathbf{v}_i^{(2)}$) with the "bottom part" of each secret key ($\mathbf{e}_{j_i}^{(2)}$). Thus, instead of only being an encryption of a function of "top part" of the secret key, the ciphertext we receive back from the KDM-CPA challenger will in fact be an encryption of f_β, which is exactly what we need.

Concretely, to respond to key-dependent message queries $(f_0 = \langle \mathbf{v}_0, \mathbf{e}_{k_0}\rangle + w_0, f_1 = \langle \mathbf{v}_1, \mathbf{e}_{k_1}\rangle + w_1, j)$, we first let $\mathbf{v}'_i = \mathbf{v}_i^{(1)}$, $w'_i = w_i + \langle \mathbf{v}_i^{(2)}, \mathbf{e}_{k_i}^{(2)}\rangle$ for $i \in \{0,1\}$. We then query the KDM-CPA challenger with $(f'_0 = \langle \mathbf{v}'_0, \mathbf{e}_{k_0}\rangle + w'_0, f'_1 = \langle \mathbf{v}'_1, \mathbf{e}_{k_1}\rangle + w'_1, j)$. Since for $i \in \{0,1\}$, $\langle \mathbf{v}'_i, \mathbf{e}_{k_i}^{(1)}\rangle + \mathbf{w}' = \langle \mathbf{v}_i, \mathbf{e}_{k_i}\rangle + \mathbf{w} = f_i$, we will receive back an encryption of $\mu = f_i$ for $i = \beta$ as

$$\mathbf{c}'^t = [\mathbf{b}_0^t \mid b_2^{(1)} + \mu],$$

where $\mathbf{b}_0^t = \mathbf{s}^t\mathbf{A}_i^* + \mathbf{x}_0^t$, $b_2^{(1)} = \mathbf{s}^t(\mathbf{y}_i^*)^{(1)} + x_2$ for $\mathbf{s} \leftarrow D_{\mathbb{Z},r}^n$, $\mathbf{x}_0 \leftarrow D_{\mathbb{Z},r}^{md}$ and $x_2 \leftarrow D_{\mathbb{Z},r}$, and $\mu = p \cdot f_\beta = p \cdot (\langle \mathbf{v}_\beta, \mathbf{e}_{j_\beta}\rangle + w_\beta)$, as in the actual scheme from Section 4.

We then compute $\mathbf{b}_1^t = -\mathbf{b}_0^t\mathbf{R} + \mathbf{x}_1^t$ and $b_2^{(2)} = \mathbf{b}_0^t\mathbf{R}\mathbf{e}_i^{(2)}$, where as in the actual scheme, $\mathbf{x}_1 \leftarrow D_{\mathbb{Z},\gamma}^w$. Letting $b_2 = b_2^{(1)} + b_2^{(2)}$, we finally output as our response

$$\mathbf{c}^t = [\mathbf{b}_0^t \mid \mathbf{b}_1^t \mid b_2 + \mu].$$

To complete the proof, we need to show that these ciphertexts are statistically indistinguishable from the ciphertexts in Game 1. The left-most part ($\mathbf{b}_0^t$) is generated by the KDM-CPA scheme with exactly the same distribution we used to generate $\mathbf{b}_0^t = \mathbf{s}^t\mathbf{A}_i^* + \mathbf{x}_0^t$ in Game 1.

In Game 1, the center part of the ciphertexts was generated as $-\mathbf{s}^t\mathbf{A}_i^*\mathbf{R} + \mathbf{x}_1^t$, while in this game, the center part is generated as $\mathbf{b}_1^t = -\mathbf{b}_0^t\mathbf{R} + \mathbf{x}_1^t = -\mathbf{s}^t\mathbf{A}_i^*\mathbf{R} + \mathbf{x}_1^t + \mathbf{x}_0^t\mathbf{R}$. Lemma 2.1 gives that $\|\mathbf{x}_0^t\mathbf{R}\| \leq \mathrm{poly}(n)$ (except with negligible probability), so that by Lemma 2.2, the statistical distance between $\mathbf{x}_1^t$ and $\mathbf{x}_1^t + \mathbf{x}_0^t\mathbf{R}$ is at most $\mathrm{poly}(n)/\gamma = \mathrm{negl}(n)$. Since the rest of the center part is distributed identically in Games 1 and 2, we have that the center part in Game 1 is statistically indistinguishable from the center part in Game 2.

Finally, in Game 1, the right part of the ciphertexts was generated as $\mathbf{s}^t\mathbf{y}_i^* + x_2 + p \cdot (\langle \mathbf{v}_\beta, \mathbf{e}_{j_\beta}\rangle + w_\beta)$. In this game, the right part is generated as $b_2 + \mu = \mathbf{s}^t(\mathbf{y}_i^*)^{(1)} + x_2 + \mathbf{s}^t\mathbf{A}_i\mathbf{R}\mathbf{e}_i^{(2)} + \mathbf{x}_0^t\mathbf{R}\mathbf{e}_i^{(2)} + \mu$, so $b_2 + \mu = \mathbf{s}^t\mathbf{y}_i^* + \mathbf{x}_0^t\mathbf{R}\mathbf{e}_i^{(2)} + x_2 + p \cdot (\langle \mathbf{v}_\beta, \mathbf{e}_{j_\beta}\rangle + w_\beta)$. Now, we have by Lemma 2.1 that $|\mathbf{x}_0^t\mathbf{R}\mathbf{e}_i^{(2)}| \leq \mathrm{poly}(n)$ except with negligible probability. Lemma 2.2 then gives that the statistical distance between x_2 and $x_2 + \mathbf{x}_0^t\mathbf{R}\mathbf{e}_i^{(2)}$ to be at most $\mathrm{poly}(n)/\gamma = \mathrm{negl}(n)$, and since the rest of the right part is distributed identically in Games 1 and 2, we have that the right part in Game 1 is statistically indistinguishable from the right part in Game 2.

Therefore, the ciphertexts in Game 1 are statistically indistinguishable from the ciphertexts in Game 2, and so the two games as a whole are statistically indistinguishable. The theorem follows. □

Acknowledgments. We thanks Oded Regev for helpful comments, and for pointing out a subtle error in a prior version of our reduction from Section 3.

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572. 2010.

[ABHS05] P. Adão, G. Bana, J. Herzog, and A. Scedrov. Soundness of formal encryption in the presence of key-cycles. In *ESORICS*, pages 374–396. 2005.

[ACPS09] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618. 2009.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

[App11] B. Applebaum. Key-dependent message security: Generic amplification and completeness. In *EUROCRYPT*, pages 527–546. 2011.

[BCHK07] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.

[BF01] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. Preliminary version in CRYPTO 2001.

[BG10] Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO*, pages 1–20. 2010.

[BGK11] Z. Brakerski, S. Goldwasser, and Y. T. Kalai. Black-box circular-secure encryption beyond affine functions. In *TCC*, pages 201–218. 2011.

[BHHI10] B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In *EUROCRYPT*, pages 423–444. 2010.

[BHHO08] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In *CRYPTO*, pages 108–125. 2008.

[BRS02] J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography*, pages 62–75. 2002.

[BV11] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *CRYPTO*, pages 505–524. 2011.

[CHK03] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. *J. Cryptology*, 20(3):265–294, 2007. Preliminary version in EUROCRYPT 2003.

[CL01] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, pages 93–118. 2001.

[CS06] S. Chatterjee and P. Sarkar. Generalization of the selective-ID security model for HIBE protocols. In *Public Key Cryptography*, pages 241–256. 2006.

[DF94] Y. Desmedt and Y. Frankel. Perfect homomorphic zero-knowledge threshold schemes over any finite abelian group. *SIAM J. Discrete Math.*, 7(4):667–679, 1994.

[DGK$^+$10] Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, pages 361–381. 2010.

[Feh98] S. Fehr. *Span Programs over Rings and How to Share a Secret from a Module*. Master's thesis, ETH Zurich, Institute for Theoretical Computer Science, 1998.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. 2009.

[GKPV10] S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*, pages 230–240. 2010.

[GM82] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. Preliminary version in STOC 1982.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. 2008.

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

[HLOV11] B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *ASIACRYPT*. 2011.

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA*, pages 319–339. 2011.

[MM11] D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *CRYPTO*, pages 465–484. 2011.

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*. 2012.

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[MR09] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post Quantum Cryptography*, pages 147–191. Springer, February 2009.

[MTY11] T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with KDM security. In *EUROCRYPT*, pages 507–526. 2011.

[OPW11] A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *CRYPTO*, pages 525–542. 2011.

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342. 2009.

[Pei10] C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *CRYPTO*, pages 80–97. 2010.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005.

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

[Ver11] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices, January 2011. Available at `http://www-personal.umich.edu/~romanv/papers/non-asymptotic-rmt-plain.pdf`, last accessed 4 Feb 2011.