Better Key Sizes (and Attacks) for LWE-Based Encryption

Richard Lindner* Chris Peikert†

November 30, 2010

Abstract

We analyze the concrete security and key sizes of theoretically sound lattice-based encryption schemes based on the "learning with errors" (LWE) problem. Our main contributions are: (1) a new lattice attack on LWE that combines basis reduction with an enumeration algorithm admitting a time/success tradeoff, which performs better than the simple distinguishing attack considered in prior analyses; (2) concrete parameters and security estimates for an LWE-based cryptosystem that is more compact and efficient than the well-known schemes from the literature. Our new key sizes are up to 10 times smaller than prior examples, while providing even stronger concrete security levels.

Keywords: lattice-based cryptography, basis reduction, learning with errors

1 Introduction

Recent years have seen significant progress in theoretically sound lattice-based cryptography, resulting in solutions to many tasks of wide applicability. In the realm of encryption alone, for example, we now have public-key cryptosystems [AD97, Reg03, Reg05] with chosen-ciphertext security [PW08, Pei09], identity-based encryption [GPV08, CHKP10, ABB10], and a fully homomorphic cryptosystem [Gen09]. Much of this progress has been greatly aided by the use of simple and flexible average-case problems — namely, the *short integer solution* (SIS) introduced by Ajtai [Ajt96] and the *learning with errors* (LWE) problem of Regev [Reg05] — that are provably as hard as certain lattice problems in the *worst case*, and appear to require time exponential in the main security parameter to solve.

For *practical* parameters, however, the concrete hardness of the SIS and LWE problems against algorithmic attacks is still far from a settled issue. This makes it difficult to assess the actual security and efficiency of cryptographic schemes that are based on these problems. The purpose of this paper is to shed further light on this issue, by considering new variants of known schemes and attacks, and analyzing their consequences in terms of key sizes and estimated security.

*Technische Universität Darmstadt. Email: rlindner@cdc.informatik.tu-darmstadt.de. This work was supported by CASED (www.cased.de).

†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. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1.1 Our Contributions

We analyze the concrete security and efficiency of modern lattice-based cryptographic schemes, with a focus on LWE and public-key encryption. To start, we describe an LWE-based cryptosystem that has substantially smaller keys and ciphertexts than the more well-known systems in the literature (namely, the original system of Regev [Reg05] and its more efficient amortized variants [PVW08, GPV08]). Our scheme incorporates several techniques and perspectives from recent works; in particular, it is an instance of an abstract system described by Micciancio [Mic10] that generalizes all the schemes of [Reg05, PVW08, GPV08], and the system's design and security proof (under the LWE assumption) combine a variety of techniques from recent works [Ale03, MR09, LPS10, Pei10] to yield asymptotic and concrete improvements in key size. While there are not any new techniques involved, to our knowledge the literature lacks a full description and analysis of the system, despite it now being an important target of study.

Our second main contribution is a new and stronger way of using existing algorithmic attack tools, such as lattice basis reduction and bounded-distance decoding with preprocessing, to analyze the concrete security of recent lattice-based cryptosystems. Our attack is directed specifically at the LWE problem, and exploits some of its structural properties in ways that have not been attempted before in a cryptanalytic context. (Our attack also does not seem immediately applicable to other lattice problems, such as the unique shortest vector problem, that have been used for public-key encryption [AD97, Reg03, AD07].) Therefore, we believe that our analysis gives a more accurate assessment of LWE's concrete hardness than estimates derived from prior lattice attacks.

Applying our attack to the improved cryptosystem, we then propose concrete parameters and (conservative) runtime estimates for modern commodity hardware. Despite our improved attacks, the resulting key sizes are still smaller than prior example parameters by factors as large as 10, even for stronger security levels. (See Section 6 for full details.) For example, using parameters that can encrypt a 128-bit payload and appear to be at least as secure as AES-128, we obtain public key sizes of about $1,120$ kilobits, or about 400 kilobits assuming a public source of trusted randomness.

Clearly, the above key sizes are still too large for many applications, but this is a consequence of the quadratic overhead inherent to the use "standard" LWE. By using the compact "ring-based" variant of LWE and cryptosystem from [LPR10] (which is related to the heuristic NTRU scheme [HPS98] and the theoretically sound line of works initiated in [Mic02]), we can immediately shrink the above key sizes by a factor of at least 200. The resulting sizes of 2-5 kilobits are comparable to modern recommendations for RSA, and the cryptosystem itself is many times faster on modern hardware.

Our methodology. Here we briefly summarize our methods and main conclusions. Our approach involves a dedicated study of basis reduction for a certain family of random lattices, and a post-reduction decoding algorithm that to our knowledge have not been considered in prior analyses. (For a discussion of our approach in relation to prior works, see Section 1.2.)

Lattice-based cryptosystems in the line of works started by Ajtai [Ajt96] involve a family of so-called *q-ary lattices*, which are m-dimensional integer lattices that contain $q\mathbb{Z}^m$ as a sublattice, for some modulus $q \geq 2$. We study how basis reduction performs, in terms of its running time and the global properties of its output basis, on random lattices from this family. Our experiments yield reliable and theoretically well-behaved predictions about the basis quality that may be obtained using various amounts of computational effort.

Complementing our analysis of lattice basis reduction, we describe a new post-reduction attack on the *search* version of the LWE problem, and provide precise trade-offs between time and adversarial advantage (i.e., success probability) in terms of the given basis quality. Even though we attack the search-LWE problem,

which is not strictly necessary to break the semantic security of most LWE-based cryptosystems, our full attack turns out to be strictly preferable (for a very wide range of parameters used in cryptography) to the natural distinguishing attack on *decision*-LWE that has been considered in prior analyses [MR09, RS10]. Specifically, our attack can solve a search-LWE instance, and hence decrypt a ciphertext, with the same or better advantage than the distinguishing attack, while using lattice vectors of lower quality and hence much less total runtime. The improvement is especially pronounced in the high-advantage regime, where the adversary needs relatively high confidence in the decrypted plaintext, such as might be required for breaking hybrid encryption.

Our post-reduction attack involves a simple extension of Babai's "nearest-plane" algorithm [Bab85] that allows us to trade basis quality against decoding time, which to our knowledge has not been explored in a cryptanalytic context. The extension is related to Klein's (de)randomized algorithm [Kle00] for bounded-distant decoding, but is simpler and specifically tailored to the known Gaussian distribution of the error vector. As we have already indicated, the quality/time trade-off dramatically affects the quality of basis required to solve an LWE instance, and hence the running time of the attack.

Finally, we note that our analysis is entirely modular, and allows for substituting improved basis reduction algorithms (and their accompanying runtime and quality predictions) into the post-reduction attack.

1.2 Related Work

Several papers contain studies of the concrete hardness of lattice problems. Here we mention the ones most closely related to our work, which are aimed at calculating secure parameters for lattice-based cryptosystems, and describe the most important distinctions.

Gama and Nguyen [GN08] performed a comprehensive study of the behavior of basis reduction for various families of lattices. Their analysis is primarily focused on the best obtainable solutions to the Hermite-, Unique-, and Approximate-Shortest Vector Problems. The Hermite SVP is in particular an important problem in our work and other cryptanalyses. While Gama and Nguyen did not attempt to document the behavior of basis reduction on random q-ary lattices (aside from the closely related Goldstein-Mayer distribution for enormous q), our experiments confirmed several of their findings for this family (as did the experiments in [MR09]). Gama and Nguyen's study was aimed mainly at predicting the behavior of basis reduction, but did not include runtime predictions, nor did it investigate the *use* of a reduced basis to solve bounded-distance decoding problems (such as LWE), where additional algorithmic ideas and trade-offs are possible.

The survey by Micciancio and Regev [MR09] proposed example parameters for various lattice-based schemes from the contemporary literature (which have larger keys than the one we describe here). Their parameters were derived using Gama and Nguyen's conclusions about the (in)feasibility of obtaining various Hermite factors, and as such do not include concrete estimates of attack runtimes or success probabilities. Their security estimates are calculated using the natural distinguishing attack on LWE by finding one relatively short vector in an associated lattice; our attack succeeds with lower-quality vectors, making it even more effective. (It should be noted that the example parameters given in [MR09] were already known to offer moderate security at best.)

Rückert and Schneider [RS10] recently gave concrete estimates of "symmetric bit security" for many recent lattice-based schemes, incorporating concrete runtime estimates for various Hermite factors in random q-ary lattices. Their analysis uses a permissive form of the distinguishing attack described in [MR09], in which the adversarial advantage is about 2^{-72}. This small advantage is not incorporated into their final bit security estimates, so the estimates are more conservative than ours, even without taking into account the superior decoding attack on search-LWE.

Finally, we note that the best distinguishing attack against LWE used in [MR09, RS10] may not always apply to our cryptosystem, because its parameters can be set so that relatively few LWE samples are published, and thus the attack is forced to use a suboptimal lattice dimension. We give further details in Sections 5.1 and 6.

2 Preliminaries

For a positive integer k, we use $[k]$ to denote the set $\{1, \ldots, k\}$. The base-2 logarithm is denoted $\lg$. We use bold lower-case letters (e.g., $\mathbf{x}$) to denote vectors over the real $\mathbb{R}$. We use bold upper-case letters (e.g., $\mathbf{B}$) for ordered sets of vectors, and identify the set with the matrix having the vectors as its columns. We let $\|\mathbf{B}\| := \max_i \|\mathbf{b}_i\|$, where $\|\cdot\|$ denotes the Euclidean norm.

For an (ordered) set of linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \subset \mathbb{R}^n$, its *Gram-Schmidt orthogonalization* $\widetilde{\mathbf{B}}$ is defined iteratively as $\widetilde{\mathbf{b}_1} = \mathbf{b}_1$, and $\widetilde{\mathbf{b}_i}$ is the component of $\mathbf{b}_i$ orthogonal to $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$ for $i = 2, \ldots, k$. In matrix notation, it corresponds to the (unique) decomposition $\mathbf{B} = \mathbf{QR}$, where the columns of $\mathbf{Q} \in \mathbb{R}^{n \times k}$ are orthonormal (i.e., $\mathbf{Q}^t\mathbf{Q} = \mathbf{I}$) and $\mathbf{R} \in \mathbb{R}^{k \times k}$ is right-triangular with positive diagonal entries; the Gram-Schmidt vectors are then $\widetilde{\mathbf{b}_i} = \mathbf{q}_i \cdot r_{i,i}$. For a set of linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$, its *fundamental parallelepiped* is

$$\mathcal{P}_{1/2}(\mathbf{B}) := \mathbf{B} \cdot [-\tfrac{1}{2}, \tfrac{1}{2})^k = \left\{ \sum_{i \in [k]} c_i \cdot \mathbf{b}_i : c_i \in [-\tfrac{1}{2}, \tfrac{1}{2}) \right\}.$$

A *lattice* Λ in $\mathbb{R}^m$ is a discrete additive subgroup. In this work we are concerned only with *q-ary* integer lattices, which are contained in $\mathbb{Z}^m$ and contain $q\mathbb{Z}^m$, i.e., $q\mathbb{Z}^m \subseteq \Lambda \subseteq \mathbb{Z}^m$. Such a lattice is generated by a (non-unique) *basis* $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_m\} \subset \mathbb{Z}^m$ of linearly independent integer vectors, as

$$\Lambda = \mathcal{L}(\mathbf{B}) := \mathbf{B} \cdot \mathbb{Z}^m = \left\{ \sum_{i \in [m]} z_i \cdot \mathbf{b}_i : z_i \in \mathbb{Z} \right\}.$$

The determinant $\det(\Lambda)$ of such a lattice is its index as a subgroup of $\mathbb{Z}^m$, i.e., $\det(\Lambda) = |\mathbb{Z}^m : \Lambda|$. Equivalently, it is $|\det(\mathbf{B})|$ for any basis $\mathbf{B}$ of Λ.

2.1 Discrete Gaussians

For a lattice Λ and a positive real $s > 0$, the *discrete Gaussian* distribution $D_{\Lambda,s}$ over Λ with parameter s is the probability distribution having support Λ that assigns a probability proportional to $\exp(-\pi\|\mathbf{x}\|^2/s^2)$ to each $\mathbf{x} \in \Lambda$. For $\Lambda = \mathbb{Z}^n$, it is easy to see (by orthonormality of its standard basis) that the discrete Gaussian $D_{\mathbb{Z}^n,s}$ is simply the product distribution of n independent copies of $D_{\mathbb{Z},s}$. There are efficient algorithms for sampling from a distribution within negligible statistical distance of $D_{\mathbb{Z},s}$, given any $s > 0$. (See, e.g., [GPV08]: for arbitrary s there is a rejection sampling algorithm, and for small s one can compute a close approximation to the cumulative distribution function.).

We will need two tail bounds on discrete Gaussians.

Lemma 2.1 ([Ban93, Lemma 1.5])**.** *Let $c \geq 1$ and $C = c \cdot \exp(\frac{1-c^2}{2}) < 1$. Then for any real $s > 0$ and any integer $n \geq 1$, we have*

$$\Pr\left[\|D_{\mathbb{Z}^n,s}\| \geq c \cdot \tfrac{1}{\sqrt{2\pi}} \cdot s\sqrt{n}\right] \leq C^n.$$

Lemma 2.2 ([Ban95, Lemma 2.4])**.** *For any real $s > 0$ and $T > 0$, and any $\mathbf{x} \in \mathbb{R}^n$, we have*

$$\Pr\left[|\langle \mathbf{x}, D_{\mathbb{Z}^n,s} \rangle| \geq T \cdot s\|\mathbf{x}\|\right] < 2\exp(-\pi \cdot T^2).$$

2.2 Learning with Errors

The *learning with errors* (LWE) problem was introduced by Regev [Reg05] as a generalization of the well-known 'learning parity with noise' problem, to larger moduli. The problem is parameterized by a dimension $n \geq 1$ and an integer modulus $q \geq 2$, as well as an error distribution χ over $\mathbb{Z}$ (or its induced distribution over $\mathbb{Z}_q$). In this work we will be concerned only with discrete Gaussian error distributions $\chi = D_{\mathbb{Z},s}$ over the integers, where $\alpha := s/q \in (0,1)$ is often called the (relative) *error rate*.

For an $\mathbf{s} \in \mathbb{Z}_q^n$, the LWE distribution $A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing a uniformly random $\mathbf{a} \in \mathbb{Z}_q^n$ and error term $e \leftarrow \chi$, and outputting the pair $(\mathbf{a}, t = \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The *search* version of the LWE problem is, given any desired number of independent samples $(\mathbf{a}_i, t_i) \leftarrow A_{\mathbf{s},\chi}$, to find $\mathbf{s}$. The *decision* version of LWE is to distinguish, with non-negligible advantage, between any desired number of independent samples $(\mathbf{a}_i, t_i) \leftarrow A_{\mathbf{s},\chi}$ (for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$), and the same number of independent samples drawn from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$. It is often convenient to write these problems in matrix form as follows: collecting the vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ as the columns of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and the (implicit) error terms $e_i \in \mathbb{Z}$ and values $t_i \in \mathbb{Z}_q$ as the entries of vectors $\mathbf{e} \in \mathbb{Z}^m$, $\mathbf{t} \in \mathbb{Z}_q^m$ respectively, we are given the input

$$\mathbf{A}, \quad \mathbf{t} = \mathbf{A}^t \mathbf{s} + \mathbf{e} \bmod q$$

and are asked to find $\mathbf{s}$, or to distinguish the input from a uniformly random $(\mathbf{A}, \mathbf{t})$. The LWE problem may also be viewed as an average-case 'bounded-distance decoding' problem on a certain family of lattices: for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, define the lattice

$$\Lambda(\mathbf{A}^t) = \{\mathbf{z} \in \mathbb{Z}^m : \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ such that } \mathbf{z} = \mathbf{A}^t \mathbf{s} \bmod q\}.$$

Then the $\mathbf{t}$ component of the LWE input may be seen as a perturbed lattice point in $\Lambda(\mathbf{A}^t)$, to be decoded.

Hardness of LWE. We recall several facts from the literature about the provable hardness of LWE. The first is that for error distribution $\chi = D_{\mathbb{Z},\alpha \cdot q}$ where $\alpha \cdot q \geq 2\sqrt{n}$, the search version of LWE is at least as hard as *quantumly* approximating certain worst-case problems on n-dimensional lattices to within $\tilde{O}(n/\alpha)$ factors [Reg05].[1] Moreover, for similar parameters and large enough q, search-LWE is at least as hard as *classically* approximating the decision shortest vector problem and variants [Pei09]. For moduli q that are sufficiently 'smooth' (i.e., products of small enough primes), the decision form of LWE is at least as hard as the search form [Reg05, Pei09].

A particularly important fact for our purposes is that decision-LWE becomes no easier to solve even if the secret $\mathbf{s}$ is chosen from the error distribution χ, rather than uniformly at random [MR09, ACPS09]. This may be seen as follows: given access to $A_{\mathbf{s},\chi}$, we can draw many samples to obtain

$$\mathbf{A}^t = \begin{bmatrix} \mathbf{A}_1^t \\ \mathbf{A}_2^t \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1^t \\ \mathbf{A}_2^t \end{bmatrix} \mathbf{s} + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix} = \mathbf{A}^t \mathbf{s} + \mathbf{e} \bmod q,$$

where $\mathbf{A}_2$ is uniform, $\mathbf{e}$ is drawn from χ, and $\mathbf{A}_1 \in \mathbb{Z}_q^{n \times n}$ is *square* and *invertible*. (This follows by forming $\mathbf{A}_1$ by greedily drawing samples that can form an invertible matrix, and disposing of any others until $\mathbf{A}_1$ is complete.) We can then transform $\mathbf{A}$ and $\mathbf{t}$ into

$$\bar{\mathbf{A}}^t := -\mathbf{A}_2^t \cdot \mathbf{A}_1^{-t} \bmod q, \quad \bar{\mathbf{t}} := \bar{\mathbf{A}}^t \mathbf{t}_1 + \mathbf{t}_2 = \bar{\mathbf{A}}^t \mathbf{e}_1 + \mathbf{e}_2 \bmod q,$$

[1] It is important to note that the original hardness result of [Reg05] is for a *continuous* Gaussian error distribution, which when rounded naively to the nearest integer does not produce a true discrete Gaussian. Fortunately, a suitable randomized rounding method does so [Pei10].

where $\bar{\mathbf{A}}$ is uniform; therefore, we have effectively replaced $\mathbf{s}$ with the error vector $\mathbf{e}_1$. On the other hand, note that when $\mathbf{A}, \mathbf{t}$ are uniformly random, then so are $\bar{\mathbf{A}}, \bar{\mathbf{t}}$.

In terms of lattices, the above may be interpreted as follows: using the bijection $\mathbf{s} \mapsto \mathbf{A}_1^t \mathbf{s}$ from $\mathbb{Z}_q^n$ to itself, we can see that the lattice $\Lambda(\mathbf{A}^t)$ defined above has as a basis the matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \\ q\mathbf{I} & -\bar{\mathbf{A}}^t \end{bmatrix}.$$

(This basis $\mathbf{H}$ is a canonical representation of $\Lambda(\mathbf{A}^t)$ known as the Hermite normal form. We have ordered the basis vectors so that the Gram-Schmidt vectors of $\mathbf{H}$ are integer multiples of the standard basis vectors, where the first several have length q, and the remainder have length 1.) Because $\mathbf{A}^t\mathbf{s} \bmod q \in \Lambda(\mathbf{A}^t)$, we have $\mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e} = \mathbf{e} \bmod \mathbf{H}$, which is

$$\mathbf{e} - \mathbf{H}\mathbf{e}_1 = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_2 + \bar{\mathbf{A}}^t\mathbf{e}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{t}} \end{bmatrix} \bmod q.$$

In conclusion, $\bar{\mathbf{t}} = \bar{\mathbf{A}}^t\mathbf{e}_1 + \mathbf{e}_2$ is the unique canonical representative of $\mathbf{e}$ modulo the lattice $\Lambda(\mathbf{A}^t)$.

Finally, assuming hardness of decision-LWE, a standard hybrid argument over the columns of $\mathbf{E}$ (see, e.g., [PW08]) shows that $(\bar{\mathbf{A}}, \bar{\mathbf{A}}^t\mathbf{E}_1 + \mathbf{E}_2)$ is indistinguishable from uniform, where the entries of $\mathbf{E} = \left[\begin{smallmatrix} \mathbf{E}_1 \\ \mathbf{E}_2 \end{smallmatrix}\right]$ are chosen independently from χ.

3 LWE-Based Encryption

Here we describe an LWE-based cryptosystem that is more space-efficient than the ones commonly known in the literature. It is an instance of an abstract system described by Micciancio [Mic10] that generalizes all the schemes of [Reg05, PVW08, GPV08], though a full description and analysis of the generalized system has not appeared in the literature. The security proof combines a number of techniques and perspectives from recent works [MR09, LPS10, Pei10] for the purpose of improved efficiency and a tight analysis. For completeness, we also briefly describe an efficient ring-based analogue of the system, which is described in full generality in the full version of [LPR10].

Despite being a generalization of prior LWE-based cryptosystems, the present scheme can actually be instantiated to have keys and ciphertexts that are smaller by a factor of about $\lg q$, while simultaneously *improving* the concrete security! The improved security comes from the smaller keys (for given security parameter n), which allows for a relatively larger noise rate that makes the LWE problem harder. The smaller keys come from a different style of security proof, which is very similar to the proofs for the coding-based cryptosystem of Alekhnovich [Ale03] and the subset sum-based cryptosystem of Lyubashevsky, Palacio, and Segev [LPS10]. In brief, the proof uses the LWE assumption *twice* (first on the public key, and then again on the ciphertext) to show that the adversary's view in a passive attack is indistinguishable from uniformly random. By contrast, the proofs for prior LWE-based schemes involve a statistical argument on either the public key or ciphertext, but this requires larger keys. We point out that statistical arguments still appear necessary for many advanced applications of LWE, such as identity-based encryption [GPV08] and others that use a 'trapdoor basis,' and we do not know whether comparably small keys and ciphertexts can be obtained for these schemes.

3.1 Cryptosystem

The cryptosystem involves a few parameters: an integer modulus $q \geq 2$ and integer dimensions $n_1, n_2 \geq 1$, which relate to the underlying LWE problems; Gaussian parameters s_k and s_e for key generation and

encryption, respectively; and a message alphabet Σ (for example, $\Sigma = \{0, 1\}$) and message length $\ell \geq 1$.

We also require a simple error-tolerant encoder and decoder, given by functions $\mathsf{encode}\colon \Sigma \to \mathbb{Z}_q$ and $\mathsf{decode}\colon \mathbb{Z}_q \to \Sigma$, such that for some large enough threshold $t \geq 1$, $\mathsf{decode}(\mathsf{encode}(m) + e \bmod q) = m$ for any integer $e \in [-t, t)$. For example, if $\Sigma = \{0, 1\}$, then we can define $\mathsf{encode}(m) := m \cdot \lfloor \frac{q}{2} \rfloor$, and $\mathsf{decode}(\bar{m}) := 0$ if $\bar{m} \in \left[-\lfloor \frac{q}{4} \rfloor, \lfloor \frac{q}{4} \rfloor\right) \subset \mathbb{Z}_q$, and 1 otherwise. This method has error tolerance $t = \lfloor \frac{q}{4} \rfloor$. We also extend encode and decode to vectors, component-wise.

To get the smallest public keys, our system makes use of a uniformly random public matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n_1 \times n_2}$ that is generated by a trusted source, and is used by all parties in the system. If there is no trusted source, then $\bar{\mathbf{A}}$ may be chosen by the user herself as part of key generation, and included in the public key.

- $\mathsf{Gen}(\bar{\mathbf{A}}, 1^\ell)$: choose $\mathbf{R}_1 \leftarrow D_{\mathbb{Z}, s_k}^{n_1 \times \ell}$ and $\mathbf{R}_2 \leftarrow D_{\mathbb{Z}, s_k}^{n_2 \times \ell}$, and let $\mathbf{P} = \mathbf{R}_1 - \bar{\mathbf{A}} \cdot \mathbf{R}_2 \in \mathbb{Z}_q^{n_1 \times \ell}$. The public key is $\mathbf{P}$ (and $\bar{\mathbf{A}}$, if needed), and the secret key is $\mathbf{R}_2$.

  In matrix form, the relationship between the public and secret keys is:

$$\begin{bmatrix} \bar{\mathbf{A}} & \mathbf{P} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix} = \mathbf{R}_1 \bmod q. \tag{3.1}$$

- $\mathsf{Enc}(\bar{\mathbf{A}}, \mathbf{P}, \mathbf{m} \in \Sigma^\ell)$: choose $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in \mathbb{Z}^{n_1} \times \mathbb{Z}^{n_2} \times \mathbb{Z}^\ell$ with each entry drawn independently from $D_{\mathbb{Z}, s_e}$. Let $\bar{\mathbf{m}} = \mathsf{encode}(\mathbf{m}) \in \mathbb{Z}_q^\ell$, and compute the ciphertext

$$\mathbf{c}^t = \begin{bmatrix} \mathbf{c}_1^t & \mathbf{c}_2^t \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1^t & \mathbf{e}_2^t & \mathbf{e}_3^t + \bar{\mathbf{m}}^t \end{bmatrix} \cdot \begin{bmatrix} \bar{\mathbf{A}} & \mathbf{P} \\ \mathbf{I} & \\ & \mathbf{I} \end{bmatrix} \in \mathbb{Z}_q^{1 \times (n_2 + \ell)}. \tag{3.2}$$

  (Note that the first ciphertext component $\mathbf{c}_1^t$ can be precomputed before $\mathbf{P}$ and $\mathbf{m}$ are known.)

- $\mathsf{Dec}(\mathbf{c}^t = [\mathbf{c}_1^t, \mathbf{c}_2^t], \mathbf{R}_2)$: output $\mathsf{decode}(\mathbf{c}_1^t \cdot \mathbf{R}_2 + \mathbf{c}_2^t)^t \in \Sigma^\ell$.

  Using Equation (3.2) followed by Equation (3.1), we are applying decode to

$$\begin{bmatrix} \mathbf{c}_1^t & \mathbf{c}_2^t \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix} = (\mathbf{e}^t + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \bar{\mathbf{m}}^t \end{bmatrix}) \cdot \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix} = \mathbf{e}^t \cdot \mathbf{R} + \bar{\mathbf{m}}^t,$$

  where $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix}$. Therefore, decryption will be correct as long as each $|\langle \mathbf{e}, \mathbf{r}_j \rangle| < t$, the error threshold of decode. (We give a formal analysis in Section 3.2 below.)

For another perspective on this scheme as an (approximate) key-agreement mechanism, let $\ell = 1$ for simplicity. By the discussion in Section 2.2, we can interpret key generation as reducing a Gaussian error vector $\mathbf{r}$ modulo a lattice defined by $\bar{\mathbf{A}}$, and publishing the result $\bar{\mathbf{A}}\mathbf{r}_2 - \mathbf{r}_1 \bmod q$. Likewise, we can view encryption as reducing a Gaussian error vector $\mathbf{e}$ modulo the *dual* of the same lattice, and publishing the result $\mathbf{e}_1^t \bar{\mathbf{A}} + \mathbf{e}_2^t \bmod q$. Using their respective private error vectors and the other party's public message, the sender and receiver can both (approximately) compute $\mathbf{e}_1^t \bar{\mathbf{A}} \mathbf{r}_2 \in \mathbb{Z}_q$, whereas a passive adversary cannot. A formal proof of security appears below in Section 3.3.

Ring-based analogue. We briefly describe a very similar scheme that is based on the decision *ring*-LWE problem [LPR10]. For messages of length any $\ell \leq n = n_1 = n_2$, and using the same values of n and q as above, the public and secret keys are up to an n factor smaller than in the above system, namely $n \lg q$ or $2n \lg q$ bits at most, depending on the availability of a common trusted string. (The ciphertext size is the same, namely $2n \lg q$ bits.)

Let $R = \mathbb{Z}[x]/f(x)$ be a polynomial ring for some monic polynomial $f(x)$ that is irreducible over $\mathbb{Z}$; common choices include *cyclotomic* polynomials such as $f(x) = x^n + 1$ for n a power of 2. (See [LPR10] for efficiency and security properties of this and other cyclotomic polynomials, including degrees n that are not powers of 2.) Let $q \in \mathbb{Z}$ be a sufficiently large integer modulus for which $f(x)$ splits into linear (or very low-degree) factors modulo q, and let $R_q = R/q = \mathbb{Z}_q[x]/f(x)$. Let χ_k, χ_e be error distributions over R that are concentrated on 'small' elements of R; see [LPR10] for what error distributions enable rigorous security proofs.

Let Σ be a message alphabet. The message encoder and decoder are functions $\mathsf{encode}\colon \Sigma^n \to R_q$ and $\mathsf{decode}\colon R_q \to \Sigma^n$, such that $\mathsf{decode}(\mathsf{encode}(m) + e \bmod q) = m$ for any 'small enough' $e \in R$, e.g., one whose coefficients as a polynomial in $\mathbb{Z}[x]/f(x)$ are all in $[-t, t)$ for some integer threshold $t \geq 1$.

As above, the system uses a uniformly random $a \in R_q$ that can be generated by a trusted source, or chosen by the user.

- $\mathsf{Gen}(a)$: choose $r_1, r_2 \leftarrow \chi_k$, and let $p = r_1 - a \cdot r_2 \in R_q$. The public key is p (and a, if needed), and the secret key is r_2.
- $\mathsf{Enc}(a, p, m \in \Sigma^n)$: choose $e_1, e_2, e_3 \leftarrow \chi_e$. Let $\bar{m} = \mathsf{encode}(m) \in R_q$, and compute the ciphertext $[c_1 = a \cdot e_1 + e_2,\ c_2 = p \cdot e_1 + e_3 + \bar{m}] \in R_q^2$.
- $\mathsf{Dec}(c = [c_1, c_2], r_2)$: output $\mathsf{decode}(c_1 \cdot r_2 + c_2) \in \Sigma^n$. By a straightforward calculation, decryption will be correct as long as $e_1 \cdot r_1 + e_2 \cdot r_2 + e_3$ is within the error threshold of decode; this holds with high probability when χ_k, χ_e are sufficiently concentrated.

The proof of security, under the decision ring-LWE assumption for noise distributions χ_k and χ_e, is essentially identical to the proof of Theorem 3.2.

3.2 Parameters for Correctness

Here we give an upper bound on the Gaussian parameters s_k, s_e in terms of the desired per-symbol error probability δ. For reasonably small values of δ, correctness for the entire message can effectively be guaranteed by way of a simple error-correcting code.

One small subtlety is that if a portion of the random vector $\mathbf{e}$ used for encryption happens to be 'too long,' then the probability of decryption error for every symbol can be unacceptably large. We address this by giving a bound on $\mathbf{e}$, in Equation (3.4) below, which is violated with probability at most $2^{-\kappa}$ for some statistical parameter κ (say, $\kappa = 40$ for concreteness). We then calculate the error probabilities assuming that the bound holds; the overall decryption error probability is then no more than $2^{-\kappa}$ larger. One can also modify the Enc algorithm to reject and resample any $\mathbf{e}$ that violates Equation (3.4); the adversary's advantage can increase by at most $2^{-\kappa}$.

Lemma 3.1 (Correctness)**.** *In the cryptosystem from Section 3.1, the error probability per symbol (over the choice of secret key) is bounded from above by any desired $\delta > 0$, as long as*

$$s_k \cdot s_e \leq \frac{\sqrt{2}\pi}{c} \cdot \frac{t}{\sqrt{(n_1 + n_2) \cdot \ln(2/\delta)}}. \tag{3.3}$$

(n_1+n_2)	$c \geq$	$(s_k \cdot s_e)/t \leq$
256	1.35	0.08936
384	1.28	0.07695
512	1.25	0.06824
640	1.22	0.06253

Figure 1: Bounds on parameters for Lemma 3.1 using a per-symbol error probability of $\delta = 0.01$, where c is determined so that the probability of choosing a 'bad' encryption vector $\mathbf{e}$ is at most 2^{-40}.

Here $c \geq 1$ is a value that depends (essentially) only on $n_1 + n_2$; representative values are given in Figure 1.

Proof. As shown above in the specification of the decryption algorithm, the jth symbol of the message decrypts correctly if $|\langle \mathbf{e}, \mathbf{r}_j \rangle| < \lfloor \frac{q}{4} \rfloor$. Recall that the entries of $\mathbf{e} \in \mathbb{Z}^{n_1+n_2+\ell}$ are independent and have distribution $D_{\mathbb{Z},s_e}$, and $\mathbf{r}_j \in \mathbb{Z}^{n_1+n_2+\ell}$ is the jth column of $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{I} \end{bmatrix}$, where the entries of $\mathbf{R}_1$ and $\mathbf{R}_2$ are drawn independently from $D_{\mathbb{Z},s_k}$.

To bound the error probability, let $\bar{\mathbf{e}} \in \mathbb{Z}^{n_1+n_2}$ consist of the first n_1+n_2 entries of $\mathbf{e}$. Then by Lemma 2.1, there is a $c \geq 1$ such that

$$\|\bar{\mathbf{e}}\| \leq c \cdot \frac{1}{\sqrt{2\pi}} \cdot s_e \sqrt{n_1 + n_2} \tag{3.4}$$

except with very small probability (concrete values of c are given in Figure 1). For any fixed $\bar{\mathbf{e}}$ satisfying the above bound, observe that each $\langle \mathbf{e}, \mathbf{r}_j \rangle$ is independent and distributed essentially as $\langle \bar{\mathbf{e}}, D_{\mathbb{Z},s_k}^{n_1+n_2} \rangle$.[2] By Lemma 2.2, for any $T \geq 0$ we have

$$\Pr\left[\left|\langle \bar{\mathbf{e}}, D_{\mathbb{Z},s_k}^{n_1+n_2} \rangle\right| \geq T \cdot s_k \|\bar{\mathbf{e}}\|\right] < 2\exp(-\pi \cdot T^2).$$

Letting $T = t/(s_k\|\bar{\mathbf{e}}\|)$, where t is the error tolerance of our message encoding, and using the bound on $\|\bar{\mathbf{e}}\|$ from above, we get the bound on $s_k \cdot s_e$ from the lemma statement. □

3.3 Security Proof

Theorem 3.2. *The cryptosystem from Section 3.1 is CPA-secure, assuming the hardness of decision-*LWE *with modulus q for: (i) dimension n_2 with error distribution $D_{\mathbb{Z},s_k}$, and (ii) dimension n_1 with error $D_{\mathbb{Z},s_e}$.*

Proof. It suffices to show that the entire view of the adversary in an IND-CPA attack is computationally indistinguishable from uniformly random, for any encrypted message $\mathbf{m} \in \Sigma^\ell$. The view consists of $(\bar{\mathbf{A}}, \mathbf{P}, \mathbf{c})$, where $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n_1 \times n_2}$ is uniformly random, $\mathbf{P} \leftarrow \mathsf{Gen}(\bar{\mathbf{A}}, 1^\ell)$, and $\mathbf{c}^t \leftarrow \mathsf{Enc}(\bar{\mathbf{A}}, \mathbf{P}, \mathbf{m})$. First, $(\bar{\mathbf{A}}, \mathbf{P})$ is computationally indistinguishable from uniformly random $(\bar{\mathbf{A}}, \mathbf{P}^*) \in \mathbb{Z}_q^{n_1 \times (n_2+\ell)}$ under assumption (i) in the lemma statement, because $\mathbf{P} = (\bar{\mathbf{A}}^t)^t \cdot (-\mathbf{R}_2) + \mathbf{R}_1$, and $\bar{\mathbf{A}}^t$ is uniform while the entries of both $-\mathbf{R}_2$ and $\mathbf{R}_1$ are drawn from $D_{\mathbb{Z},s_k}$. So the adversary's view is indistinguishable from $(\mathbf{A}, \mathbf{c})$ where $\mathbf{A} = (\bar{\mathbf{A}}, \mathbf{P}^*)$ is uniformly random and $\mathbf{c} \leftarrow \mathsf{Enc}(\mathbf{A}, \mathbf{m})$. Now $(\mathbf{A}, \mathbf{c})$ is also computationally indistinguishable from uniformly random $(\mathbf{A}, \mathbf{c}^*)$ under assumption (ii) in the lemma statement, because $\mathbf{c} = (\mathbf{A}^t\mathbf{e}_1 + \begin{bmatrix} \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix}) + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}$, and $\mathbf{A}$ is uniform while the entries of $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are drawn from $D_{\mathbb{Z},s_e}$. □

[2] We ignore the one additional term drawn from $D_{\mathbb{Z},s_e}$, which is compensated for by some slack in our final choice of parameters.

It should be noted that for some settings of the parameters, one of the two assumptions in Theorem 3.2 may be true *information-theoretically* for the number of LWE samples exposed by the system in an attack. For instance, if $n_2 \geq n_1 \lg q$ and $s_k \geq \omega(\sqrt{\log n_1})$, then the public key $(\bar{\mathbf{A}}, \mathbf{P})$ is within a negligible (in n_1) statistical distance of uniformly random (by a suitable version of the leftover hash lemma), whereas the corresponding ciphertexts are statistically far from uniform. These properties are important in, for example, the 'dual' cryptosystem and identity-based encryption scheme of [GPV08]. Conversely, the applications found in [PVW08, BHY09, ACPS09] have public keys that are far from uniform, but require that encryption under a 'malformed' (uniformly random) public key produces a ciphertext that is statistically independent of the encrypted message. These properties are achieved when $n_1 \geq n_2 \lg q$ and $s_e \geq \omega(\sqrt{\log n_2})$, again by the leftover hash lemma.

4 Lattice Decoding Attacks

The most promising practical attacks on the cryptosystem from Section 3, and more generally on LWE itself, use lattice-basis reduction followed by a decoding phase using the reduced basis.[3] In this section we analyze the performance of decoding as it relates to the quality of a given reduced basis. Then in Section 5 we analyze the effort required to obtain bases of a desired quality.

Before proceeding, we briefly explain how our *decoding* attack on LWE differs from the *distinguishing* attacks considered in other works [MR09, RS10]. In the latter, the adversary distinguishes (with some noticeable advantage) an LWE instance $(\mathbf{A}, \mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e})$ from uniformly random, which is typically enough to break the semantic security of an LWE-based cryptosystem with the same advantage. To do this, the adversary finds a short nonzero integral vector $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = \mathbf{0} \bmod q$, which may be seen as a short vector in the (scaled) dual of the LWE lattice $\Lambda(\mathbf{A}^t)$. (Equivalently, the points of $\Lambda(\mathbf{A}^t)$ may be partitioned into hyperplanes orthogonal to $\mathbf{v}$, successively separated by distance $q/\|\mathbf{v}\|$.) The adversary then simply tests whether the inner product $\langle \mathbf{v}, \mathbf{t} \rangle$ is "close" to zero modulo q. When $\mathbf{t}$ is uniform, the test accepts with probability exactly $1/2$, but when $\mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e}$ for Gaussian $\mathbf{e}$ with parameter s, we have $\langle \mathbf{v}, \mathbf{t} \rangle = \langle \mathbf{v}, \mathbf{e} \rangle \bmod q$, which is essentially a Gaussian (reduced mod q) with parameter $\|\mathbf{v}\| \cdot s$. When this parameter is not much larger than q, the Gaussian (mod q) can be distinguished from uniform with advantage very close to $\exp(-\pi \cdot (\|\mathbf{v}\| \cdot s/q)^2)$. For example, when $\|\mathbf{v}\| = 4q/s$ the distinguishing advantage is about 2^{-72}. However, to distinguish (and hence decrypt a ciphertext) with high confidence, one needs $\|\mathbf{v}\| \leq q/(2s)$ or so, which usually requires a great deal more effort to obtain.

It is customary to include the inverse distinguishing advantage in the total 'cost' of an attack, so the computational effort and advantage need to be carefully balanced. For practical parameters, the optimal total cost of the distinguishing attack typically involves a very small distinguishing advantage (see Section 6), which may not be very useful in some settings, such as hybrid encryption.

Our decoding attack is stronger than the distinguishing attack in that it can actually recover the secret error vector in the LWE instance (and hence decrypt the ciphertext) with the same or better advantage, while using lower-quality vectors. For all the parameter settings that we investigated, our attack yields a better total effort as a ratio of time/advantage, and it is significantly more efficient in the high-advantage regime. (See Section 6 and Figure 4 in particular for details.) The attack works by using an entire reduced basis (not just one vector), and by expending some additional post-reduction effort to find the LWE solution. We also point out that unlike in basis reduction, the post-reduction effort is fully parallelizable.

[3] There are also purely combinatorial attacks on LWE [BKW03, Wag02] that may perform asymptotically better than lattice reduction, but so far not in practice. Also, these attacks generally require more LWE samples than our cryptosystem exposes, and an exponentially large amount of space.

The attack. Recall from Section 2.2 that an LWE instance $(\mathbf{A}, \mathbf{t} = \mathbf{A}^t\mathbf{s} + \mathbf{e})$ may be seen as a bounded-distance decoding problem on a certain lattice $\Lambda = \Lambda(\mathbf{A}^t)$, where $\mathbf{A}^t\mathbf{s} \in \Lambda$.

The standard method for solving a bounded-distance decoding problem on lattices is the recursive NearestPlane algorithm of Babai [Bab85]. The input to the algorithm is some lattice basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ (which for best results should be as reduced as possible) and a target point $\mathbf{t} \in \mathbb{R}^m$, and the output is a lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ that is 'relatively close' to $\mathbf{t}$. The precise guarantee is that for any $\mathbf{t} \in \mathrm{span}(\mathbf{B})$, $\mathsf{NearestPlane}(\mathbf{B}, \mathbf{t})$ returns the unique $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\mathbf{t} \in \mathbf{v} + \mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$. In other words, if $\mathbf{t} = \mathbf{v} + \mathbf{e}$ for some $\mathbf{v} \in \mathcal{L}(\mathbf{B})$, the algorithm outputs $\mathbf{v}$ if and only if $\mathbf{e}$ happens to lie in $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$.

The main drawback of this approach in attacking LWE is that in a reduced basis $\mathbf{B}$, the last several Gram-Schmidt vectors of $\mathbf{B}$ are typically very short, whereas the first few are relatively long. In such a case, the parallelepiped $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$ is very 'long and skinny,' and so the Gaussian error vector $\mathbf{e}$ is very unlikely to land in it, causing NearestPlane to produce an incorrect answer.

We address this issue by giving a generalized algorithm that admits a time/success tradeoff. It works just as NearestPlane does, except that it can recurse on some $d_i \geq 1$ distinct planes in the ith level of the recursion. In essence, the multiple recursion has the effect of making the parallelepiped $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$ wider in the direction of $\widetilde{\mathbf{b}_i}$ by a factor of exactly d_i.[4] To capture the most probability mass of the Gaussian error distribution of $\mathbf{e}$, one should choose the multiples d_i so as to maximize $\min_i(d_i \cdot \|\widetilde{\mathbf{b}_i}\|)$.[5]

The input to our NearestPlanes algorithm is a lattice basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \subset \mathbb{R}^m$, a vector $\mathbf{d} = (d_1, \ldots, d_k) \in (\mathbb{Z}^+)^k$ of positive integers, and a target point $\mathbf{t} \in \mathbb{R}^m$. It outputs a set of $\prod_{i \in [k]} d_i$ distinct lattice vectors in $\mathcal{L}(\mathbf{B})$, as follows:

1. If $k = 0$, return $\mathbf{0}$. Else, let $\mathbf{v}$ be the projection of $\mathbf{t}$ onto $\mathrm{span}(\mathbf{B})$.
2. Let $c_1, \ldots, c_{d_k} \in \mathbb{Z}$ be the d_k distinct integers closest to $\langle \widetilde{\mathbf{b}_k}, \mathbf{v} \rangle / \langle \widetilde{\mathbf{b}_k}, \widetilde{\mathbf{b}_k} \rangle$.
3. Return $\bigcup_{i \in [d_k]} (c_i \cdot \mathbf{b}_k + \mathsf{NearestPlanes}(\{\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}\}, (d_1, \ldots, d_{k-1}), \mathbf{v} - c_i \cdot \mathbf{b}_k))$.

Note that the recursive calls to NearestPlanes can be run entirely in parallel. The following lemma is an immediate extension of the analysis from [Bab85].

Lemma 4.1. *For* $\mathbf{t} \in \mathrm{span}(\mathbf{B})$, $\mathsf{NearestPlanes}(\mathbf{B}, \mathbf{d}, \mathbf{t})$ *returns the set of all* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *such that* $\mathbf{t} \in \mathbf{v} + \mathcal{P}_{1/2}(\widetilde{\mathbf{B}} \cdot \mathbf{D})$*, where* $\mathbf{D} = \mathrm{diag}(\mathbf{d})$*. The running time is essentially* $\prod_{i \in [k]} d_i$ *times as large as that of* $\mathsf{NearestPlane}(\mathbf{B}, \mathbf{t})$.

Note that the columns of $\widetilde{\mathbf{B}} \cdot \mathbf{D}$ from the lemma statement are the orthogonal vectors $d_i \cdot \widetilde{\mathbf{b}_i}$, so $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}} \cdot \mathbf{D})$ is a rectangular parallelepiped with axis lengths $d_i \cdot \|\widetilde{\mathbf{b}_i}\|$.

[4] The algorithm of Klein [Kle00] also can recurse on more than one plane per iteration. Klein's algorithm solves the general bounded-distance decoding problem, and selects the planes at each stage probabilistically (though it can also be derandomized); its guarantee is related solely to the shortest Gram-Schmidt vector in the basis. Our algorithm is tailored specifically to the setting where we know the distribution of the offset vector; this allows the algorithm to recurse on exactly those planes that maximize the probability of success (over the choice of the error vector).

[5] One could further generalize the algorithm to search within an approximate *ball* made up of 'bricks' that are copies of $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$, thus capturing even more of the Gaussian without adding much more to the search space. However, this would significantly complicate the analysis, and we find that the present approach is already very effective.

Success probability of NearestPlanes**.** When $\mathbf{t} = \mathbf{v} + \mathbf{e}$ for some $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ and a *continuous* Gaussian $\mathbf{e} \leftarrow D_s$ for some $s > 0$, the probability that $\mathbf{v}$ is in the output set of $\mathsf{NearestPlanes}(\mathbf{B}, \mathbf{d}, \mathbf{t})$ is

$$\Pr\left[\mathbf{e} \in \mathcal{P}_{1/2}(\widetilde{\mathbf{B}} \cdot \mathrm{diag}(\mathbf{d}))\right] = \prod_{i=1}^{m} \Pr\left[|\langle \mathbf{e}, \widetilde{\mathbf{b}_i}\rangle| < d_i \cdot \langle \widetilde{\mathbf{b}_i}, \widetilde{\mathbf{b}_i}\rangle/2\right] = \prod_{i=1}^{m} \mathrm{erf}\left(\frac{d_i \cdot \|\widetilde{\mathbf{b}_i}\|\sqrt{\pi}}{2s}\right), \quad (4.1)$$

which follows by the independence of the values $\langle \mathbf{e}, \widetilde{\mathbf{b}_i}\rangle$, due to the orthogonality of the Gram-Schmidt vectors $\widetilde{\mathbf{b}_i}$. When $\mathbf{e}$ is drawn from a sufficiently wide *discrete* Gaussian over the integer lattice (in practice, a parameter of 6 or more suffices), the above is an extremely close approximation to the true probability.

We conclude this section by giving an informal explanation for why the advantage of the decoding attack can potentially be much larger than that of the distinguishing attack above, given vectors of the same quality. In the distinguishing attack, using a vector $\mathbf{v}$ of length (say) $\|\mathbf{v}\| \approx 4q/s$ implies that $\langle \mathbf{v}, \mathbf{t}\rangle \bmod q$ is distributed roughly as D_{4q} modulo q, whose statistical distance is only about 2^{-72} from uniform. A basis $\mathbf{B}$ of $\Lambda(\mathbf{A}^t)$ of equivalent quality has $\|\widetilde{\mathbf{b}_m}\| = q/\|\mathbf{v}\| = s/4$, because $\Lambda(\mathbf{A}^t)$ lies in hyperplanes orthogonal to $\mathbf{v}$ and separated by distance $q/\|\mathbf{v}\|$. So even without using multiple recursion in NearestPlanes (i.e., letting every $d_m = 1$), the corresponding term in Equation (4.1) is $\mathrm{erf}(\sqrt{\pi}/8) \approx 0.25$; moreover, the remaining terms typically approach 1 very rapidly, since $\|\widetilde{\mathbf{b}_i}\|$ usually increases quickly as i decreases. Letting $d_i > 1$ increases the overall success probability even more at little added cost, and allows for obtaining a relatively large advantage without needing higher-quality basis vectors.

5 Basis Reduction and Experiments

In this section we present an analysis of lattice basis reduction on random q-ary lattices arising from LWE, and results of reduction experiments on various parameters. Our goal is to predict a conservative, but still useful, lower bound on the practical runtime of the lattice decoding attack described in Section 4 for a given set of LWE parameters.

We found that the best practical lattice reduction algorithm currently available to us is the BKZ algorithm as implemented by Shoup in the NTL library [Sho], so this is what we used in our experiments. The BKZ algorithm is parameterized by a blocksize k between 2 and the dimension of the lattice to be reduced. As the blocksize increases, the reduced basis improves in quality (i.e., it contains shorter lattice vectors, whose Gram-Schmidt lengths are closer together), but the runtime of BKZ also rapidly increases, becoming practically infeasible for $k \geq 30$ or so.

There has been some recent progress in the development of algorithms for finding short vectors in lattices, which can be used as subroutines to (or entire replacements of) BKZ reduction. For example, Gama, Nguyen, and Regev [GNR10] recently proposed a new method called "Extreme Enum", which is much faster than its predecessor, the Schnorr-Euchner enumeration [SE94]. There are also single-exponential time algorithms for the Shortest Vector Problem [AKS01, MV10b, MV10a], which can run faster in practice than Schnorr-Euchner enumeration in certain low dimensions; however, these algorithms also require exponential space. We were not able to evaluate the performance and effectiveness of all these approaches, leaving this for future work. The BKZ implementation we use employs Schnorr-Euchner enumeration and, since the BKZ framework uses the enumeration subroutine as a black box, we presume that new algorithms incorporating Extreme Enum and other approaches will soon be available for evaluation. (For a comparison of enumeration algorithms in practice, see the open SVP-challenge website.[6])

[6] `http://www.latticechallenge.org/svp-challenge/`

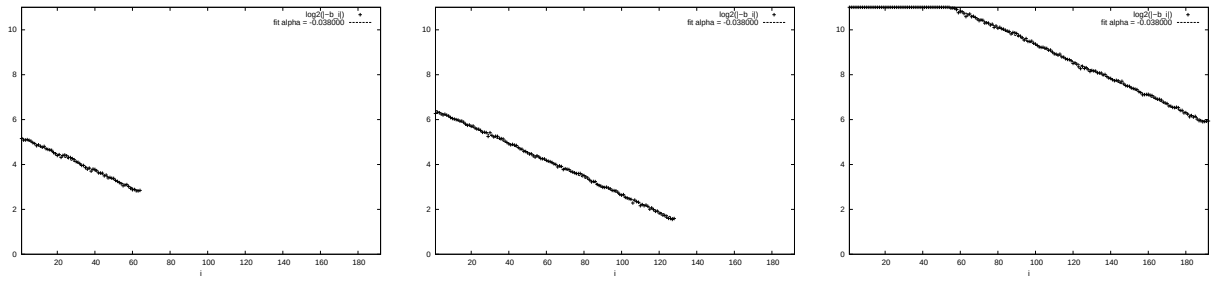

Figure 2: Logarithmic GSO lengths of three LWE instances after BKZ-20 reduction, which conform to the GSA assumption (modified with fixed upper and lower bounds on the Gram-Schmidt lengths). In all cases, the observed slope of the plot is very nearly the same, but other parameters vary. Parameters are $n = 32, q = 257, m = 64$ (left); $n = 64, q = 257, m = 128$ (center); $n = 32, q = 2053, m = 192$ (right).

In Section 5.1, we analyze the main properties of BKZ-reduced bases for q-ary lattices that are relevant to our decoding attack. In Section 5.2, we use our experiments to estimate the runtime required to obtain bases of a desired quality. We point out that the rest of our analysis is independent of this estimate, and can easily be applied with other runtime estimates for BKZ variants or other approaches.

5.1 Basis Reduction for q-ary Lattices

We begin by reviewing some of the prior work on basis reduction, in particular as applied to the q-ary lattices that arise from LWE.

The analysis of lattice reduction algorithms by Gama and Nguyen [GN08] identified the *Hermite factor* of the reduced basis as the dominant parameter in the runtime of the reduction and the quality of the reduced basis. A basis $\mathbf{B}$ of an m-dimensional lattice Λ has Hermite factor δ^m for $\delta \geq 1$ if $\|\mathbf{b}_1\| = \delta^m \cdot \det(\Lambda)^{1/m}$. For convenience, we call δ the *root-Hermite factor*.

Another important concept is the *Geometric Series Assumption* (GSA), introduced by Schnorr [Sch03]. The GSA says that in a BKZ-reduced basis $\mathbf{B}$, the lengths $\|\widetilde{\mathbf{b}_i}\|$ of the Gram-Schmidt vectors decay geometrically with i, namely, $\|\widetilde{\mathbf{b}_i}\| = \|\mathbf{b}_1\| \cdot \alpha^{i-1}$ for some $0 < \alpha < 1$. Our experiments on random q-ary lattices adhere to the GSA very closely, with the exception that the Gram-Schmidt lengths are always upper- and lower-bounded by q and 1 respectively, owing to the special structure of q-ary lattices (see Figure 2). For large BKZ blocksizes that correspond to effective attacks on LWE, these exceptional cases do not arise, and our bases conform to the GSA as ordinarily stated.

By combining the notion of Hermite factor with the GSA, we can predict the lengths of *all* Gram-Schmidt vectors in a basis $\mathbf{B}$ (of an m-dimensional lattice Λ) having root-Hermite factor δ. An easy calculation shows that under the GSA,

$$\det(\Lambda) = \prod_{i=1}^{m} \|\widetilde{\mathbf{b}_i}\| = \alpha^{m(m-1)/2} \cdot \delta^{m^2} \cdot \det(\Lambda) \quad \Longrightarrow \quad \alpha = \delta^{-2m/(m-1)} \approx \delta^{-2}, \tag{5.1}$$

where the approximation holds for large m.

We now turn to q-ary lattices that arise from LWE. Recall from Section 2.2 that LWE is a bounded-distance decoding problem on the m-dimensional lattice

$$\Lambda(\mathbf{A}^t) = \{\mathbf{z} \in \mathbb{Z}^m : \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ such that } \mathbf{z} = \mathbf{A}^t\mathbf{s} \bmod q\}$$

for some $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with $m \geq n$. Because the LWE problem allows us to ignore some of the rows of $\mathbf{A}^t$ (and the corresponding noisy inner products), a natural and important question is what 'subdimension' m makes a lattice attack most effective. This question was addressed in [MR09], where a simple calculation showed that for a desired root-Hermite factor δ, the subdimension $m = \sqrt{n \lg(q) / \lg(\delta)}$ is optimal in the context of the natural distinguishing attack on LWE (as described at the beginning of Section 4). The analysis of [MR09] actually applies to the lattice

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\},$$

which is the *dual* of $\Lambda(\mathbf{A}^t)$ up to scaling by a q factor, and the optimal subdimension m given above minimizes the length of $\widetilde{\mathbf{d}_1} = \mathbf{d}_1$ in a reduced basis $\mathbf{D}$ of $\Lambda^{\perp}(\mathbf{A})$ having root-Hermite factor δ. In our setting, by duality the same choice of m *maximizes* $\|\widetilde{\mathbf{b}_m}\| = q/\|\widetilde{\mathbf{d}_1}\|$, where the basis $\mathbf{B}$ of $\Lambda(\mathbf{A}^t)$ is the dual basis of $\mathbf{D}$ in *reverse* order.

In our decoding attack (and assuming the GSA), the form of the success probability given in Equation (4.1) as a product of $\mathrm{erf}(\cdot)$ terms also strongly indicates that we should maximize $\|\widetilde{\mathbf{b}_m}\|$, and hence use the same subdimension $m = \sqrt{n \lg(q) / \lg(\delta)}$ as above. We do not have a fully rigorous proof of this claim, since using a smaller m decreases the number of terms in the product, and hence could potentially increase the success probability. However, it seems unlikely that using a smaller m would improve the success probability by much (if at all). This is because $\|\widetilde{\mathbf{b}_m}\| = q/\|\mathbf{d}_1\|$ decreases rapidly as m decreases (see [MR09]), and $\|\widetilde{\mathbf{b}_{m-i}}\| \approx \|\widetilde{\mathbf{b}_m}\| \cdot \delta^{2(i-1)}$ is a very close approximation for small i, which are the Gram-Schmidt vectors that largely determine the success probability. Likewise, increasing m also appears counterproductive, since it both decreases $\|\widetilde{\mathbf{b}_m}\|$ *and* increases the number of terms in the product.

All of the above assumes that a cryptosystem exposes enough LWE samples (via its public keys and/or ciphertexts) to use the optimal subdimension. While this is always true of prior cryptosystems [Reg05, PVW08, GPV08], it is not necessarily the case for our cryptosystem in Section 3, due to its smaller keys and ciphertexts. In this case, the adversary should use the dimension m corresponding to the actual number of published samples (this rule applies to some of our parameters sets given in Section 6).

5.2 Extrapolating BKZ Runtimes

In order to assign concrete runtimes to the attacks we put forward, we need to predict the runtime required to achieve a given root-Hermite factor δ in random q-ary lattices.

Gama and Nguyen [GN08] observed that on random lattices generated according to a variety of models, the runtime required to achieve a given root-Hermite factor δ in large dimensions (exceeding 200 or so) is largely determined by δ alone; the lattice dimension and determinant contribute only second-order terms. Our initial experiments confirmed this behavior for random q-ary lattices, and so we extrapolated runtimes using a fixed set of LWE parameters q and n, for a variety of values δ that correspond to sufficiently large optimal subdimensions $m = \sqrt{n \lg(q) / \lg(\delta)} \approx 200$. Our experiments were performed on a single 2.3 GHz AMD Opteron machine, using the single-precision floating-point BKZ implementation from the standard NTL library [Sho]. (Practical attacks on LWE for parameters beyond toy examples would require using at least quadruple precision, which would increase the running times by at least some constant factor, so our extrapolations are somewhat optimistic and hence conservative from a security point of view.)

Figure 3 shows the results of our experiments and their extrapolations. Using the rule of thumb that obtaining a 2^k approximation to the shortest vector in an m-dimensional lattice takes time $2^{\tilde{O}(m/k)}$ using BKZ, we conclude that the logarithm of the runtime should grow roughly linearly in $1/\lg(\delta)$. Our limited experiments seem consistent with this behavior, though many more would be needed to confirm it with

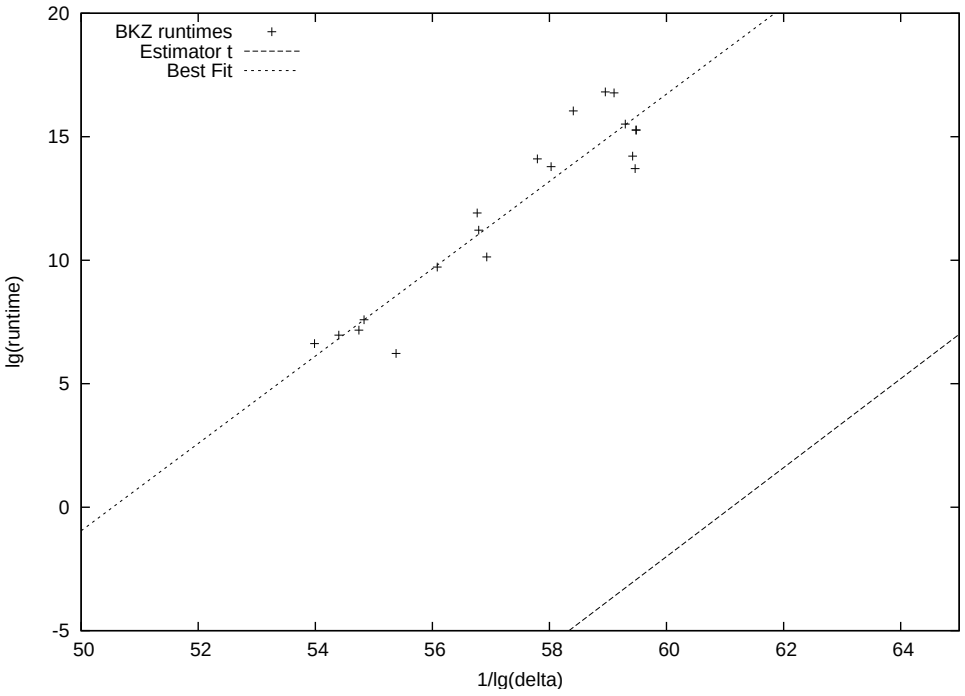

Figure 3: Runtime of BKZ experiments on random q-ary lattices, with parameters $n = 72$, $q = 1021$, and $m = \sqrt{n \lg(q)/\lg(\delta_0)}$, i.e., the optimal subdimension with respect to a desired root-Hermite factor δ_0. The vertical axis is $t_{\mathsf{BKZ}}(\delta) := \lg(T_{\mathsf{BKZ}}(\delta))$, the logarithmic runtime required to obtain a vector with root-Hermite factor δ when running BKZ with successively increasing blocksizes. The horizontal axis is $1/\lg(\delta)$ for the actual root-Hermite factor δ achieved by the reduction. For comparison, the graph shows the best-fit estimator $t_{\mathsf{BKZ}}(\delta) = 1.086/\lg(\delta) - 91$, and our conservative lower bound estimate $t_{\mathsf{BKZ}}(\delta) = 1.8/\lg(\delta) - 110$.

confidence. Using least-square regression, the best linear fit to our data for $t_{\mathsf{BKZ}}(\delta) := \lg(T_{\mathsf{BKZ}}(\delta))$, the log runtime (in seconds, on our machine) of BKZ as a function of δ, is $t_{\mathsf{BKZ}}(\delta) = 1.806/\lg(\delta) - 91$. Since our experiments were limited by resources and available time, and we expect to see further improvements in basis reduction techniques (such as those in [GNR10]), for analyzing concrete hardness we use a conservative lower bound estimate of

$$t_{\mathsf{BKZ}}(\delta) := \lg(T_{\mathsf{BKZ}}(\delta)) = 1.8/\lg(\delta) - 110. \tag{5.2}$$

Note that in this estimate, the 1.8 factor is very slightly smaller, and the -110 constant term is substantially smaller, than their counterparts in the best-fit function from our experiments. We chose the value 1.8 because our experiments were limited to relatively small block sizes, and the runtimes needed to achieve smaller values of δ very quickly became infeasible, so we believe that the true coefficient on the linear term (even with improved algorithms) is larger than 1.8. Similarly, our choice of -110 provides for some security margin against special-purpose hardware. In conclusion, we believe that our lower bound estimate provides some safety against foreseeable advances in algorithms and hardware, but in any case, our analysis is entirely modular and can be immediately adapted to work with any revised estimator.

6 Cryptosystem Parameters

We now estimate the concrete security of, and compute the space requirements for, the LWE-based cryptosystem from Section 3 on a variety of parameters, and compare with the example parameters given in [MR09] for the cryptosystem described therein (which is essentially due to [PVW08]). Figure 4 provides the security estimates, and Figure 5 gives key and ciphertext sizes.

Instantiating the parameters. We set the cryptosystem's parameters as $n_1 = n_2 = n$ and $s_k = s_e = s$ for some positive integer n and $s > 0$, so that the two LWE hardness assumptions made in Theorem 3.2 are equivalent. In practice, though, distinguishing the public key and ciphertext from uniform are not equally hard, because the public key exposes fewer LWE samples than the ciphertext does. In particular, the adversary cannot use the optimal subdimension in attacking the public key, making it quite a bit harder to break. This fact could allow us to use slightly smaller s_k and correspondingly larger s_e parameters to get slightly stronger overall security, but we elect not to introduce such complications at this point. (And arguably, the secret key ought to be better-protected than any individual ciphertext.)

We choose the modulus q to be just large enough (according to the bounds in Figure 1) to allow for a Gaussian parameter $s \geq 8$, so that the discrete Gaussian $D_{\mathbb{Z}^m,s}$ approximates the continuous Gaussian D_s extremely well.[7] Increasing the value of q beyond this threshold appears not to increase the concrete security of our cryptosystem, and (somewhat paradoxically) may even slightly decrease it! This is because the BKZ runtime depends almost entirely on the root-Hermite factor δ, and by the constraints on our parameters (specifically, $s_k = s_e = s = O(\sqrt{q})$), the δ yielding a successful attack on our system grows as $q^{\Theta(1/n)}$, which increases with q (albeit very slowly).

Estimating the security. We analyze the distinguishing attack and our decoding attack (both described in Section 4), estimating the total runtimes for each of a few representative adversarial advantages. The attacks apply to a single key and ciphertext; by a standard hybrid argument, the advantage increases at most linearly in the number of ciphertexts encrypted under a single key.

For analyzing the basic distinguishing attack we rely on calculations from [MR09]. We first compute a bound $\beta = (q/s) \cdot \sqrt{\ln(1/\varepsilon)/\pi}$ on the length of a nonzero vector $\mathbf{v} \in \Lambda^{\perp}(\mathbf{A})$ that would yield the desired distinguishing advantage (taken over the random choice of the LWE error). We then compute the root-Hermite factor $\delta = 2^{(\lg^2 \beta)/(4n \lg q)}$ that would yield such a vector, assuming that the attacker can use the optimal subdimension $m = \sqrt{n \lg(q)/\lg(\delta)}$. (The value of δ follows from the fact that in the optimal subdimension, a root-Hermite factor of δ yields a vector of length $2^{2\sqrt{n \lg q \lg \delta}}$.) If the optimal subdimension for this δ exceeds $n_1 + n_2 + \ell = 2n + 128$ (the number of LWE samples implicitly exposed by a ciphertext), then we discard this δ and instead use the one for which $\delta^m \cdot q^{n/m} = \beta$, where $m = 2n + 128$. (Values of δ computed in this way are indicated in Figure 4 by asterisks.) We then calculate a lower bound on the BKZ runtime using our conservative estimator from Equation (5.2).

In analyzing our decoding attack, we try various values of δ, computing both the estimated BKZ runtime and the number of enumerations needed (assuming the GSA) to achieve the desired success probability according to Equation (4.1). If the number of enumerations does not exceed the BKZ runtime (in seconds) by more than a 2^{16} factor, we consider this to be an acceptable attack. (This 2^{16} factor is somewhat arbitrary, but

[7] Note that the theoretical worst-case reduction [Reg05] for LWE asks that $s \geq 2\sqrt{n}$. However, the constant factors are not tight, and here we are concerned with concrete hardness against known attacks.

			Adv. ε	(Distinguish)		(Decode)		
n	q	s	$\lg(1/\varepsilon)$	δ	$\lg(\text{secs})$	δ	$\lg(\#\text{enum})$	$\lg(\text{secs})$
			≈ 0	$^*1.0065$	83	1.0089	47	32
128	2053	6.77	-32	1.0115	< 0	1.0116	13	< 0
	(toy)		-64	1.0128	< 0	1.0130	1	< 0
			≈ 0	$^*1.0045$	168	1.0067	87	78
192	4093	8.87	-32	1.0079	49	1.0083	54	42
	(low)		-64	1.0087	34	1.0091	44	29
			≈ 0	$^*1.0034$	258	$^*1.0052$	131	132
256	4093	8.35	-32	1.0061	96	1.0063	87	90
	(medium)		-64	1.0067	77	1.0068	73	75
			≈ 0	$^*1.0027$	353	$^*1.0042$	163	189
320	4093	8.00	-32	1.0049	146	1.0052	138	132
	(high)		-64	1.0054	122	1.0055	117	119
			≈ 0	1.0038	219	1.0071	82	68
136	2003	13.01	-32	1.0088	33	1.0092	42	27
	[MR09]		-64	1.0098	18	1.0102	27	14
			≈ 0	1.0053	126	1.0078	66	52
214	16381	7.37	-32	1.0091	28	1.0094	39	25
	[MR09]		-64	1.0099	17	1.0102	29	14

Figure 4: Example parameters and attacks for the LWE-based cryptosystem described in Section 3.1, for various adversarial advantages. The cryptosystem parameters are $n = n_1 = n_2$, q, $s = s_k = s_e$, and message length $\ell = 128$ bits. For comparison, the last two parameter settings ($n = 136$, $n = 214$) come from the example parameters of [MR09]. The columns labelled "Distinguish" refer to a distinguishing (i.e., semantic security) attack. These give the root-Hermite factors δ needed to obtain the respective distinguishing advantages (over the random choice of the LWE error vector), and the corresponding logarithmic runtime (in seconds) according to our optimistic estimator from Equation (5.2). The columns labelled "Decode" refer to our decoding (i.e., message and randomness recovery) attack. These give example root-Hermite factors and number of NearestPlanes enumerations needed to obtain the respective decoding probability, and the corresponding estimated runtime of the attack. Other trade-offs between δ and the number of enumerations are possible (as δ increases, so does #enum); we chose the largest δ for which the estimated enumeration runtime does not exceed that of basis reduction. *An asterisk on a value of δ indicates that for reduced vectors of lengths required by the attack, the cryptosystem reveals too few LWE samples to allow an optimal choice of subdimension and corresponding root-Hermite factor δ. In such cases, we used the value of δ induced by working with the full dimension $m = n_1 + n_2 + \ell = 2n + 128$.

n	q	s	Per-User Key ($\mathbf{P}$)	Full Key ($\mathbf{P}$ & $\bar{\mathbf{A}}$)	Ciphertext ($\mathbf{c}$)	Msg Expansion
128	2053	6.77	1.8×10^5	3.6×10^5	2.8×10^3	22.0
192	4093	8.87	2.9×10^5	7.4×10^5	3.8×10^3	30.0
256	4093	8.35	4.0×10^5	11.2×10^5	4.6×10^3	36.0
320	4093	8.00	4.9×10^5	17.2×10^5	5.4×10^3	42.0
136	2003	13.01	2.8×10^6	5.8×10^6	2.9×10^3	22.6
214	16381	7.37	2.4×10^6	6.4×10^6	4.8×10^3	18.7

Figure 5: Sizes (in bits) of public keys and ciphertexts for the cryptosystem described in Section 3; for comparison, the last two rows are for parameters given in [MR09]. In each case, the message size is $\ell = 128$ bits. The "message expansion" factor is the ratio of ciphertext size to plaintext size. Recall that in the ring-based system, the public key sizes are about a factor of n smaller.

seems to be a reasonable estimate on the number of NearestPlanes enumerations that can be performed per second, especially with parallelism.) We list the largest value of δ for which we found an acceptable attack, along with the corresponding runtime (which includes both the BKZ and NearestPlanes phases).

Conclusions. We highlight a few notable conclusions from our analysis:

1. The decoding attack is always at least as good as the distinguishing attack for all reasonable advantages, and is vastly superior in the high-advantage regime. As an extreme example, decoding is more than 2^{160} times faster when obtaining a large advantage against the "high security" ($n = 320$) parameter set.
2. For the "low security" ($n = 192$) parameter set, our key sizes are about 10 times smaller than those in [MR09], while offering somewhat better security, and the "high security" parameters are approximately 4-5 times smaller. Note also that the ring-based scheme has key sizes about a factor of n smaller again.
3. For the "medium security" ($n = 256$) parameter set, the best runtime/advantage ratio is approximately 2^{120} seconds, which translates on our machine to about 2^{150} operations. It seems reasonable to conclude that these parameters currently offer security levels at least matching those of AES-128. While we elect not to give precise "symmetric bit security" claims owing to the approximate nature of our predicted runtimes, rough figures could be derived using the heuristics of Lenstra and Verheul [LV01].

Acknowledgments

We thank Vadim Lyubashevsky, Markus Rückert, and Michael Schneider for helpful discussions, and for pointing out irregularities in our previous security estimates. We also thank the CT-RSA reviewers for their useful comments.

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572. 2010. On p. 1.

[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. On pp. 5 and 10.

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293. 1997. On pp. 1 and 2.

[AD07] M. Ajtai and C. Dwork. The first and fourth public-key cryptosystems with worst-case/average-case equivalence. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(97), 2007. On p. 2.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996. On pp. 1 and 2.

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610. 2001. On p. 12.

[Ale03] M. Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307. 2003. On pp. 2 and 6.

[Bab85] L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986. Preliminary version in STACS 1985. On pp. 3 and 11.

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993. On p. 4.

[Ban95] W. Banaszczyk. Inequalites for convex bodies and polar reciprocal lattices in R^n. *Discrete & Computational Geometry*, 13:217–231, 1995. On p. 4.

[BHY09] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35. 2009. On p. 10.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. On p. 10.

[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. On p. 1.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. 2009. On p. 1.

[GN08] N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, pages 31–51. 2008. On pp. 3, 13, and 14.

[GNR10] N. Gama, P. Q. Nguyen, and O. Regev. Lattice enumeration using extreme pruning. In *EUROCRYPT*, pages 257–278. 2010. On pp. 12 and 15.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. 2008. On pp. 1, 2, 4, 6, 10, and 14.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288. 1998. On p. 2.

[Kle00] P. N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, pages 937–941. 2000. On pp. 3 and 11.

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23. 2010. On pp. 2, 6, and 8.

[LPS10] V. Lyubashevsky, A. Palacio, and G. Segev. Public-key cryptographic primitives provably as secure as subset sum. In *TCC*, pages 382–400. 2010. On pp. 2 and 6.

[LV01] A. K. Lenstra and E. R. Verheul. Selecting cryptographic key sizes. *J. Cryptology*, 14(4):255–293, 2001. On p. 18.

[Mic02] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Preliminary version in FOCS 2002. On p. 2.

[Mic10] D. Micciancio. Duality in lattice cryptography. In *Public Key Cryptography*. 2010. Invited talk. On pp. 2 and 6.

[MR09] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post Quantum Cryptography*, pages 147–191. Springer, February 2009. On pp. 2, 3, 4, 5, 6, 10, 14, 16, 17, and 18.

[MV10a] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In *STOC*, pages 351–358. 2010. On p. 12.

[MV10b] D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *SODA*, pages 1468–1480. 2010. On p. 12.

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342. 2009. On pp. 1 and 5.

[Pei10] C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *CRYPTO*, pages 80–97. 2010. On pp. 2, 5, and 6.

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, pages 554–571. 2008. On pp. 2, 6, 10, 14, and 16.

[PW08] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196. 2008. On pp. 1 and 6.

[Reg03] O. Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004. Preliminary version in STOC 2003. On pp. 1 and 2.

[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. On pp. 1, 2, 5, 6, 14, and 16.

[RS10] M. Rückert and M. Schneider. Selecting secure parameters for lattice-based cryptography. Cryptology ePrint Archive, Report 2010/137, 2010. http://eprint.iacr.org/. On pp. 3, 4, and 10.

[Sch03] C.-P. Schnorr. Lattice reduction by random sampling and birthday methods. In *STACS*, pages 145–156. 2003. On p. 13.

[SE94] C.-P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathmatical Programming*, 66:181–199, 1994. On p. 12.

[Sho] V. Shoup. Number theory library 5.5.2 (NTL) for C++. http://www.shoup.net/ntl/. On pp. 12 and 14.

[Wag02] D. Wagner. A generalized birthday problem. In *CRYPTO*, pages 288–303. 2002. On p. 10.