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

Richard Lindner* Chris Peikert†

November 22, 2010

Abstract

We analyze the concrete security and associated key sizes for theoretically sound lattice-based encryption schemes based on the "learning with errors" (LWE) problem. Our main contributions are (1) a new, detailed model and experimental analysis of how basis-reduction and post-reduction attacks perform on the specific family of random lattices arising from the use of LWE, and (2) concrete parameters and security estimates for an LWE-based cryptosystem that is more compact and efficient than the more well-known schemes from the literature. For security levels exceeding that of a 128-bit symmetric cipher, our new key sizes are at least 10 times smaller than prior recommendations.

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 variants [PVW08, GPV08]). The present 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 proof of security (under the LWE assumption) combines a variety of techniques from [Ale03, MR09, LPS10, Pei10]. To our knowledge, however, the current literature lacks a full description and analysis of the system, despite it now being an important target of study.

The main contribution of this work is a new way of using known 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 attacks are directed specifically at the LWE problem, and take advantage of some of its structural properties in ways that do not seem possible against 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 precise and accurate picture of LWE's concrete hardness than estimates derived from more generic lattice attacks. We then apply our methodology to propose new concrete parameters and estimated security levels for the LWE-based cryptosystem mentioned above. Despite our improved attacks, the resulting key sizes are still smaller than prior recommendations by a factor of more than 10, for security levels exceeding that of a 128-bit symmetric cipher. Specifically, we obtain public key sizes of about 430 kilobits (for encrypting a 128-bit payload), or about 200 kilobits for per-user keys assuming a source of common trusted randomness.

Clearly, the above key sizes are still too large for many applications, but this is a consequence of the $\Omega(n^2)$ cost inherent to the use "standard" LWE. Fortunately, 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 key sizes further by a factor of about 100 or more. The resulting key sizes of 4-5 kilobits are competitive with modern recommendations for RSA, and the cryptosystem itself is many times faster on modern hardware.

Our methodology. Here we briefly summarize our approach, which involves a dedicated study of basis reduction for a certain family of random lattices, and some additional post-reduction attack strategies that to our knowledge have not been considered in prior analyses. (For a discussion of our approach in relation to prior work, 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 of determinant q^n that contain $q\mathbb{Z}^m$ as a sublattice, for some modulus $q \geq 2$ and parameter $n \leq m$. (For reasons relating to the worst-case hardness reductions, n is usually taken as the main hardness parameter.) We perform a detailed study of how basis reduction performs, in terms of its running time and the global quality of its output basis, on random lattices from this family as a function of the main parameters n, q, and m. Our experiments and analysis provide reliable and theoretically well-behaved predictions about the basis quality that may be obtained using various amounts of computational effort.

Complementing our predictions for lattice basis reduction, we then explore post-reduction attacks on the LWE problem itself, and give precise trade-offs between time and success probability in terms of a given basis quality. This involves a simple, but to our knowledge unexplored in this context, extension of Babai's "nearest-plane" decoding algorithm [Bab85] that allows us to trade basis quality against extra time. (The

extension is related to Klein's (de)randomized algorithm [Kle00] for bounded-distant decoding, but is simpler and optimally exploits the known Gaussian distribution of the offset vector.) The quality/time tradeoff can dramatically affect the optimal balance of effort between basis reduction and post-reduction decoding, and the quality of basis required to solve an LWE instance. Combining these two main pieces, we then give security estimates that reflect both time and overall success probability, and calculate practical parameters for various levels of concrete security.

We note that our analysis is quite modular and allows for, e.g., substituting other basis reduction algorithms and analyses (with runtime and/or 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 toward calculating secure parameters for lattice-based cryptosystems.

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. While these are very important problems to study in general, they are typically not the most natural ways of expressing cryptographic problems on q-ary lattices, which ask to find (or decode) lattice vectors within some fixed *absolute* distance bounds. Aside from the Goldstein-Mayer distribution for very large moduli (which are not typically used in cryptographic constructions), their work also did not attempt to document the behavior of basis reduction on q-ary lattices. More importantly, it did not investigate the *use* of a reduced basis to solve bounded-distance decoding problems (of which LWE is a special case), where additional algorithmic ideas and trade-offs are possible.

The survey by Micciancio and Regev [MR09] proposed parameters for various lattice-based schemes from the contemporary literature. Their parameters were derived using Gama and Nguyen's general estimates for the best obtainable Hermite factor, and as such do not include concrete security estimates (e.g., of symmetric bit security), nor do they incorporate specific information about q-ary lattices or post-reduction decoding. Their parameters apply to a less efficient LWE-based encryption scheme that has larger keys than the one studied here.

Rückert and Schneider [RS10] recently gave concrete estimates of 'bit security' for many recent lattice-based schemes, incorporating concrete running time and output quality predictions on q-ary lattices. Their attacks are primarily focused on the Hermite Shortest Vector Problem, which also immediately yields a distinguishing attack (with non-negligible advantage) on the LWE problem — but not necessarily the most effective one. Specifically, the post-reduction phase of our attack is able to use somewhat longer vectors, and (when successful) actually recovers the secret randomness (and hence the encrypted message) with certainty. Finally, the best attacks described 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.

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

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

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

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

[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].

(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 $\lg q$, with little or no apparent loss in concrete security. This is due to 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*

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

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

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

$(n_1 + n_2)$	$c \geq$	$(s_k \cdot s_e)/t \leq$
192	1.41	0.09879
256	1.35	0.08936
288	1.33	0.08552
320	1.31	0.08236
384	1.28	0.07695
448	1.26	0.07237
512	1.25	0.06824

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}.

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: (1) dimension n_1 and error distribution $D_{\mathbb{Z},s_e}$, and (2) dimension n_2 and error $D_{\mathbb{Z},s_k}$.*

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 (2) 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 (1) in the lemma statement, because $\mathbf{c} = (\mathbf{A}^t\mathbf{e}_1 + \left[\begin{smallmatrix}\mathbf{e}_2\\ \mathbf{e}_3\end{smallmatrix}\right]) + \left[\begin{smallmatrix}\mathbf{0}\\ \mathbf{m}\end{smallmatrix}\right]$, and $\mathbf{A}$ is uniform while the entries of $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are drawn from $D_{\mathbb{Z},s_e}$. □

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

[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.

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.

As described in Section 2.2, an LWE instance $(\mathbf{A}, \mathbf{t} = \mathbf{A}^t\mathbf{e}_1 + \mathbf{e}_2)$ may be seen as a bounded-distance decoding problem on a lattice $\Lambda = \mathcal{L}(\mathbf{H})$ having basis

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

where $\mathbf{e} = \left[\begin{smallmatrix}\mathbf{e}_1 \\ \mathbf{e}_2\end{smallmatrix}\right] = \left[\begin{smallmatrix}\mathbf{0} \\ \mathbf{t}\end{smallmatrix}\right] \bmod \mathbf{H}$. (Recall that we have ordered the columns of $\mathbf{H}$ so that its first several Gram-Schmidt vectors have length q, and the remainder have length 1.) Solving the LWE instance means finding $\mathbf{e}$, or equivalently, decoding $\mathbf{t}$ to the lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{H})$ such that $\left[\begin{smallmatrix}\mathbf{0} \\ \mathbf{t}\end{smallmatrix}\right] = \mathbf{v} + \mathbf{e}$.

Note that it is not necessary to use the full dimensionality of the above basis; one can always truncate some of the last rows and the corresponding columns of $q\mathbf{I}$. This corresponds to (temporarily) ignoring the bottom rows of $\mathbf{A}^t$ and $\mathbf{e}_2$; once $\mathbf{e}_1$ is found, $\mathbf{e}_2 = \mathbf{t} - \mathbf{A}^t\mathbf{e}_1$ can then be computed directly. Note also that as shown in the proof of Theorem 3.2, the semantic security of the cryptosystem from Section 3 can only be violated by successfully attacking the LWE instances implicit in the public key or the ciphertext.

Attacking LWE. 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 $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \subset \mathbb{R}^m$ (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 $\mathbf{t} \in \mathrm{span}(\mathbf{B})$, NearestPlane$(\mathbf{B}, \mathbf{t})$ returns the unique $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\mathbf{t} \in \mathbf{v} + \mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$.

A basic attack on search-LWE using the nearest-plane algorithm proceeds as follows. Given $(\mathbf{A}, \mathbf{t} = \mathbf{A}^t\mathbf{e}_1 + \mathbf{e}_2)$, first find a reduced basis $\mathbf{B} \in \mathbb{Z}^{m \times m}$ for $\Lambda(\mathbf{A}^t)$, starting from the non-reduced basis $\mathbf{H}$ defined above. Then run NearestPlane$(\mathbf{B}, \mathbf{t})$ in an attempt to find the error vector $\mathbf{e}$. This succeeds if and only if $\mathbf{e}$ happens to lie in $\mathcal{P}_{1/2}(\widetilde{\mathbf{B}})$.

The main drawback of this approach is that the last several Gram-Schmidt vectors of $\mathbf{B}$ are typically very short, having length very close to 1 (whereas the first few Gram-Schmidt vectors are quite long, having length about q). 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 fail.

We address this issue by describing a generalized algorithm that admits a time/success tradeoff. It works just as NearestPlane does, except that it can recurse on more than one plane per iteration. 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, the number of different multiples of $\mathbf{b}_i$ that the algorithm considers. To best 'capture' the

[3] There are other attacks on LWE [BKW03, Wag02] that may perform asymptotically better, but not in practice. Also, they generally require more LWE samples than our cryptosystem exposes, and an exponentially large amount of space.

probability mass of the Gaussian error distribution, one should choose the multiples d_i so as to maximize $\min_i(d_i \cdot \|\widetilde{\mathbf{b}_i}\|)$.[4]

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 $\bigcup_{i \in [d_k]} (c_i \cdot \mathbf{b}_k + \mathsf{NearestPlanes}(\{\mathbf{b}_1, \ldots, \mathbf{b}_{k-1}\}, (d_1, \ldots, d_{k-1}), \mathbf{v} - c_i \cdot \mathbf{b}_k)$.

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 $d_i \cdot \|\widetilde{\mathbf{b}_i}\|$.

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 $s > 0$, 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.

5 Basis Reduction and Experiments

In this section we present results from lattice basis reduction experiments on more than $5{,}000$ LWE lattice bases with various parameters. Our goal is to accurately predict the practical runtime of the lattice decoding attack described in Section 4 for a given set of LWE parameters n, q, m, and Gaussian error parameter s.

We found that the best practical lattice reduction algorithm to obtain large Gram-Schmidt vectors is the BKZ algorithm as implemented by Shoup in the NTL library [Sho], so this is the one we used in our experiments; we have derived a formula predicting its runtime for a given set of parameters. The BKZ algorithm is parameterized by a blocksize k between 2 and the dimension of the lattice to be reduced. With large blocksizes (say, 30 or more), BKZ requires enormous runtime, but the resulting Gram-Schmidt vectors are of much higher quality.

[4] There is another algorithm by Klein [Kle00], that, like ours, recurses on one or more planes, and not necessarily the ones nearest to the target. Klein's algorithm is concerned with the general bounded-distance decoding problem, and selects the planes probabilistically (though it can also be derandomized). Our algorithm is tailored specifically to LWE, where we know the distribution of the error vector from the lattice to the target. This allows our algorithm to recurse on exactly those planes that maximize the probability of success (over the choice of the error vector).

There has been some recent progress in the development of alternative algorithms that find short lattice vectors, which can be used as subroutines to (or entire replacements of) BKZ reduction. For example, Gama, Nguyen, and Regev proposed a new method called "Extreme Enum" [GNR10], 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 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. (For a practical comparison of enumeration algorithms, see the open SVP-challenge website.[5])

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 combining Extreme Enum and other approaches with the BKZ framework will soon be available for evaluation. In Section 5.4, we estimate the runtime required to run BKZ on q-ary lattices. The rest of our analysis is independent of this estimate, and can easily be applied with runtime estimates for BKZ variants or other approaches.

For our analysis, we start by introducing a slight modification to the well-known Geometric Series Assumption (GSA) heuristic, which was introduced by Schnorr [Sch03]. Using the new heuristic and our experimental data, we are then able to accurately predict the lengths of all Gram-Schmidt vectors after BKZ reduction has been performed with a given blocksize. Finally, we use these predictions to estimate runtimes and success probabilities for a full attack on LWE.

5.1 The (Bounded) Geometric Series Assumption

The Geometric Series Assumption (GSA) states that after some minimal amount of lattice basis reduction, say using LLL, the logarithmic lengths of the Gram-Schmidt vectors follow an affine linear function. In other words, there are constants $\alpha, \beta \in \mathbb{R}$ such that $\lg(\|\widetilde{\mathbf{b}}_i\|) = \alpha \cdot i + \beta$.

During our experiments, we observed that for certain bases this heuristic is true only after a very large amount of reduction work, which depends on the dimension and determinant of the input basis. This was because in large dimensions m, a very large blocksize was required before all $m - n$ of the initial length-q orthogonal basis vectors were nontrivially incorporated into the reduced basis. (We also observed complementary behavior when m was very close to n, and several length-1 GSO vectors were left unaltered by the reduction.) However, there is a slightly more general heuristic, which appears to hold true for reduced bases of q-ary lattices, and allows for further insights. We call this the *Bounded GSA assumption*. The condition is that there are constants $\alpha, \beta, \gamma \in \mathbb{R}$ such that

$$\lg(\|\widetilde{\mathbf{b}}_i\|) = \begin{cases} \gamma & \text{if } \alpha \cdot i + \beta > \gamma, \\ 0 & \text{if } \alpha \cdot i + \beta < 0, \\ \alpha \cdot i + \beta & \text{otherwise.} \end{cases}$$

Note that in the relevant case of q-ary lattices Λ of some given determinant, all the constants are determined by fixing the slope α, because $\gamma = \lg(q)$ and β is determined by the constraint $\sum_{i=1}^{m} \lg \|\widetilde{\mathbf{b}}_i\| = \lg(\det(\Lambda))$.

During our experiments, we found that the lengths predicted by the Bounded GSA model are very close to the actual GSO lengths in all cases. We also found that the slope α, which can be observed after BKZ reduction, depends almost entirely on the blocksize k and not on other parameters n, q, and m (see Figure 2).

Another observation regarding Bounded GSA is that one can predict the slope (and thus all GSO lengths) for a given blocksize, using the Gaussian heuristic to predict the length of the shortest non-zero vector in a

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

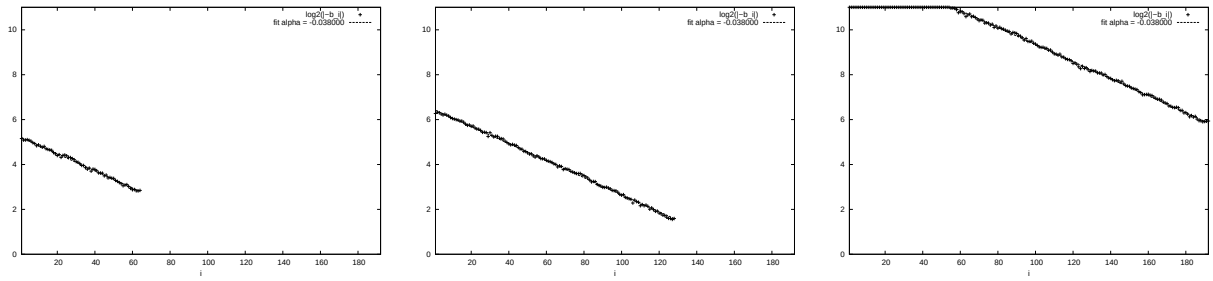

Figure 2: Logarithmic GSO lengths of three LWE instances after BKZ-20 reduction, which conform to the Bounded GSA assumption. In all cases, the observed slope is $\alpha \approx -0.038$, but other parameters vary. Parameters are $n = 32, q = 257, m = 64$ (left); $n = 64, q = 257, m = 128$ (center); $n = 32, q = 2053, m = 192$ (right).

lattice. This allows us to conservatively predict GSO slopes for huge blocksizes, for which we do not have the resources to run actual experiments.

5.2 GSO Slopes and Projected Lattices

Assuming that the Bounded GSA as described in Section 5.1 holds, we want to predict the slope (and thus all GSO lengths) achieved by a BKZ reduction for a given blocksize k. For this we will use the Gaussian heuristic to predict the length of the shortest non-zero vector in a lattice. This will allow us to conservatively predict GSO slopes for huge blocksizes, for which we do not have the resources to run actual experiments.

Let Λ be a lattice with a basis $\mathbf{B}$ that has been BKZ-reduced using blocksize k. Let $\eta(i) = \lg(\|\widetilde{\mathbf{b}}_i\|)$ be the function describing the logarithmic GSO lengths. We assume that the blocksize is large enough so that there is a k-dimensional projected lattice $\bar{\Lambda}$ with respect to $\mathbf{B}$ for which the ratio of area above and below η is the same as the ratio for the whole basis, i.e.,

$$\frac{\lg(\det(q\Lambda^*))}{\lg(\det(\Lambda))} = \frac{\lg(\det(q\bar{\Lambda}^*))}{\lg(\det(\bar{\Lambda}))} = \frac{n}{m-n}.$$

Experimentally, these projected lattices always exist and are entirely in the *sloped* part of the GSO plot for blocksizes ≥ 30, where we will use the predicted values.

First, note that we have $\lg(\det(\bar{\Lambda}))+\lg(\det(q\bar{\Lambda}^*)) = k\lg(q)$, so in fact $\lg(\det(\bar{\Lambda})) = \lg(\det(\Lambda))\cdot(k/m)$. In other words, considering the function $\bar{\eta}(i) = \lg(\|\widetilde{\mathbf{b}}_i\|)$, the values of η for the basis block corresponding to $\bar{\Lambda}$, we know that $\bar{\eta}((k+1)/2) = \lg(\det(\bar{\Lambda}))/k = \frac{m-n}{m}\cdot\lg(q)$.

We also know that $\bar{\eta}$ lies entirely in the sloped part, so $\bar{\eta}(i) = \alpha(i-(k+1)/2) + \frac{m-n}{m}\cdot\lg(q)$, for some slope α that we want to find. Now, since $\bar{\Lambda}$ is a k-dimensional projected lattice of a BKZ-reduced basis with blocksize k, we know it is in fact HKZ-reduced. So, it is reasonable to assume that the Gaussian heuristic holds for $\bar{\Lambda}$. This states that

$$\begin{aligned}\bar{\eta}(1) &= \lg(\lambda_1(\bar{\Lambda})) \\ &= \lg(\det(\bar{\Lambda}))/k - \lg(\pi)/2 + \lg(\Gamma(1+k/2))/k.\end{aligned}$$

Solving for α, we find $\alpha = \lg(\pi)/(k-1) - 2\lg(\Gamma(1+k/2))/(k(k-1))$. See Figure 3 for a plot of blocksizes and their corresponding values of α.

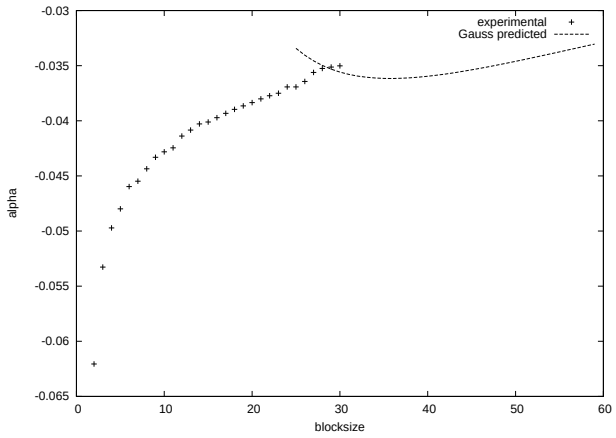

Figure 3: Slopes for the Bounded GSA for different blocksizes. For small blocksizes $k \leq 30$, the shown data is experimentally verified; for larger blocksizes the slopes are predicted using the Gaussian heuristic.

5.3 GSO Slopes and Hermite Factors

Many analyses of lattice reduction algorithms (e.g., [GN08]) consider the resulting *Hermite factor* of the reduced basis. A basis $\mathbf{B}$ for an m-dimensional lattice Λ has Hermite factor δ^m if $\|\mathbf{b}_1\| = \delta^m \det(\Lambda)^{1/m}$. Assuming the Bounded GSA, we can connect the GSO lengths to the Hermite factor as follows. For a basis $\mathbf{B}$, let $\eta(i) = \lg(\|\widetilde{\mathbf{b}}_i\|)$ be the function describing the corresponding logarithmic GSO lengths. For q-ary lattices under Bounded GSA, we have

$$\eta(i) = \alpha\left(i - \frac{m+1}{2}\right) + \frac{m-n}{m}\lg(q).$$

Now, assuming that the basis has Hermite factor δ^m, we know $\eta(1) = \lg(\|\mathbf{b}_1\|)$ and can solve for the corresponding slope α. We find the relation

$$\alpha = -\frac{2m\lg(\delta)}{m-1} \approx -2\lg(\delta). \tag{5.1}$$

So, instead of using the slopes we found in our experiments, and predicted with the Gaussian heuristic for larger block sizes, one can alternatively use the predicted Hermite factor for a given blocksize (or another basis reduction technique entirely) and derive the associated slope using Equation (5.1).

5.4 Extrapolating BKZ Runtimes

Predicting the GSO lengths will allow us to calculate success probabilities for the post-reduction run of NearestPlanes, but in order to complete our analysis, we also need to predict the runtime of a BKZ reduction for a given blocksize k and LWE parameters n, q, and m. The main parameters determining the runtime are the dimension m and the blocksize k. Like Gama and Nguyen [GN08], we observed that the logarithm of the runtime grows linearly in the lattice dimension m and quadratically in k for $k \geq 10$. We also found that the log runtime grew linearly in two other quantities: the LWE dimension n, and the logarithm of the corresponding lattice determinant $d := \lg(\det(\Lambda)) = (m-n)\lg(q)$.

The model that best fit our experimental data is as follows. Let $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3] \in \mathbb{R}^{4\times 3}$ be a matrix of constant factors to be determined later, and consider a model for the logarithm $t_{\mathsf{BKZ}} = \lg(T_{\mathsf{BKZ}})$ of the BKZ

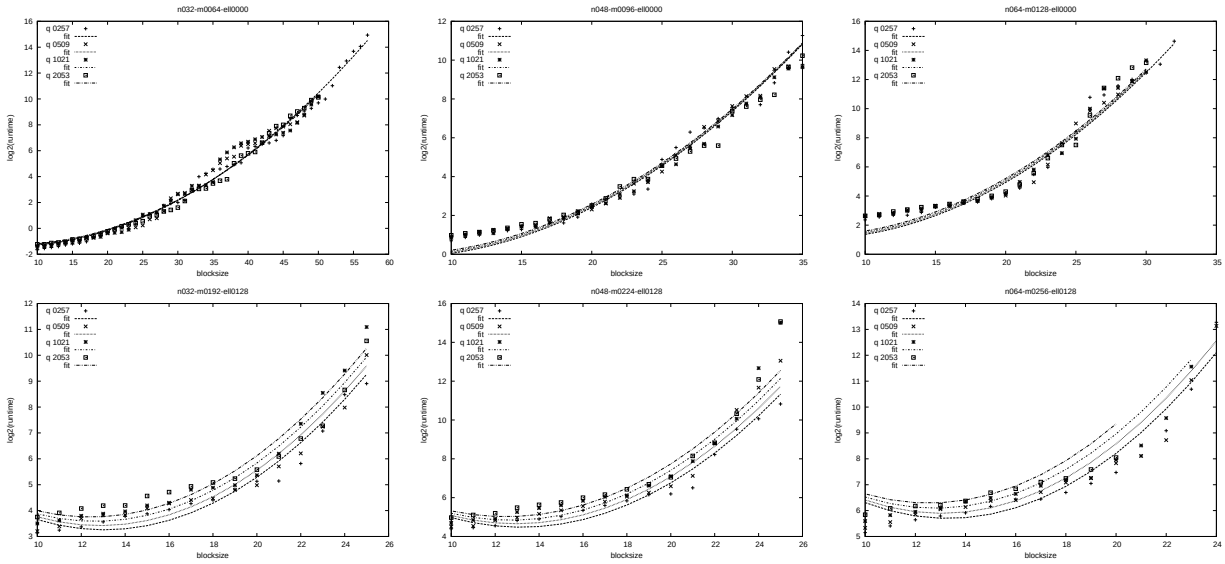

Figure 4: Logarithmic runtime of BKZ for blocksizes $k \geq 10$ on various LWE instances. Every plot shows data points for $q = 257, 509, 1021, 2053$ and the runtime predicted by our model function T. The main parameters are $n = 32$ (left); $n = 48$ (center); $n = 64$ (right). The secondary parameter is $m = 2n$ (top); $m = 2n + 128$ (bottom). Note that the lower right plot is missing several data points, because (as predicted) the corresponding reductions take too long to perform.

runtime as

$$t_{\mathsf{BKZ}}(k, n, q, m) = (m, n, d, 1) \cdot \mathbf{c}_1 \cdot (k + (m, n, d, 1) \cdot \mathbf{c}_2)^2 + (m, n, d, 1) \cdot \mathbf{c}_3. \tag{5.2}$$

We used least-square fitting to find constants $\mathbf{C}$ such that our above model for t_{BKZ} best fits the average logarithmic runtime for the reduction of 5 random LWE bases with corresponding parameters. See Figures 4 for some examples of our runtime predictions versus the real data in different settings. The values that best fit our experimental data (under least-squares fit) are

$$\mathbf{C} = \begin{pmatrix} 0.067 & -0.00030 & 0.028 \\ -0.12 & 0.00014 & 0.012 \\ -0.0013 & 0.0000018 & 0.0010 \\ 5.5 & 0.0078 & -3.8 \end{pmatrix}.$$

Note that our experiments used only *single-precision* floating point BKZ. Practical attacks on large parameters would require using at least quadruple precision, which would increase the logarithmic running times by some additive constant term.

5.5 Concrete Security Estimates

We consider the following model for a concrete attack against the LWE-based cryptosystem from Section 3. The attack is divided into two phases: a computationally heavy "preprocessing" phase in which we apply basis reduction to a public key, and an "online" phase that invests a proportionally smaller amount of effort

against each of a potentially large number of ciphertexts. For best results, the total amount of effort spent in each phase should be about the same, so the effort in preprocessing will generally be much larger than the effort against any particular ciphertext. This model is justified by the fact that a public key is frequently long-lived, and all ciphertexts encrypted under that key are relative to the same lattice. Therefore, an attacker can gain the greatest probability of breaking at least one ciphertext by investing a one-time effort to reduce the public key lattice basis as much as possible, thereby amortizing that effort over all of the ciphertexts.

One possible outcome of the preprocessing phase is that one or more of the short secret key vectors (i.e., columns of $\mathbf{R}_2$) can be exposed by decoding with NearestPlanes. This obviously constitutes a significant break, because the attacker can then decrypt the corresponding symbols of every ciphertext with essentially no additional effort. However (and unsurprisingly), our experiments show that the attacker has a greater likelihood of success if it uses its reduced basis with NearestPlanes to attempt to decode many individual ciphertexts, investing roughly an equal amount of effort on each of them. The downside (to the attacker) is that a successful attack will typically only reveal the contents of a few encrypted messages (not of its choosing), but this still constitutes a break of the scheme's semantic security.

Precise description of the attack. We now use our experimental data to estimate the concrete security of the LWE-based cryptosystem. In our instantiation, we set the 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 underlying security (in Theorem 3.2) are equivalent. We choose q to be just large enough (according to the bounds in Figure 1) to allow for a Gaussian parameter $s \geq 6$, which as we shall see appears to be sufficient for security against known attacks.[6] We will also see that increasing the value of q beyond this threshold appears not to increase the concrete hardness much at all, because the time required for BKZ reduction and the slope of the resulting logarithmic GSO lengths are essentially independent of q (all other parameters remaining the same).

Faced with the parameters n, q, and s, the attacker is left to allocate its effort between the preprocessing phase (BKZ reduction of the public key lattice) and the online phase (NearestPlanes decoding of many ciphertexts). As mentioned above, it is best to allocate roughly equal time to both phases, so the ratio of effort allocated to BKZ versus a single call to NearestPlanes should be roughly the number of ciphertexts N available to the attacker. To be conservative, we use $N \approx 2^{40}$; in certain applications, it may be reasonable to assume a smaller value of N, and this will lead to stronger estimated security levels.

For the BKZ reduction phase, the attacker must choose a blocksize k and a sublattice dimension m for BKZ reduction. To find the optimal choices for our attack model, we enumerated the following process over all blocksizes $20 \leq k \leq n$ and dimensions $n < m \leq 2n + \ell$:

1. Using the predicted BKZ runtime $T_{\mathsf{BKZ}} = 2^{t_{\mathsf{BKZ}}}$ for these parameters (Equation (5.2)), we compute the total number of recursions that can be performed by NearestPlanes in the budgeted time of T_{BKZ}/N.

2. Using the predicted GSO lengths for the BKZ-reduced basis, we estimate the success probability of each run of NearestPlanes, and the resulting probability of successfully decrypting *any* of the N ciphertexts.

3. We compute the ratio of total running time divided by success probability, treating it as a measure of "total expected runtime" on our computer.

Finally, we minimize the total expected runtime over all choices of k and m.

[6] Note that theoretical worst-case reductions [Reg05] for LWE ask that $s \geq 2\sqrt{n}$, but the constant factors are not tight.

n	q	s	k	m	lg(BKZ Time)	lg(# Recur)	lg(Succ Prob)	lg(Exp Runtime)	Symmetric Security
96	2053	7.11	29	246	25.7	0	-10.0	41.1	64 (1992)
128	2053	6.77	28	276	30.1	0	-52.7	83.8	107 (2048)
144	4093	9.35	28	308	34.8	4	-81.5	117.2	140 (2092)
160	4093	9.17	28	326	38.6	8	-122.3	161.9	185 (2150)
192	4093	8.87	28	359	47.0	16	-222.6	270.5	293 (2292)
136	2003	13.01	28	287	31.9	41	-135.0	168.0	191 (2158)
214	16381	7.37	22	642	28.0	38	-210.0	239.0	262 (2251)

Figure 5: Security estimates for our LWE-based cryptosystem, following the attack model described in Section 5.5, assuming a corpus of up to 2^{40} ciphertexts for a single public key. The LWE parameters are $n = n_1 = n_2$, q, and $s = s_k = s_e$. The best attack we found (by exhaustive search) uses blocksize k and LWE sublattices of dimension m. The estimated BKZ runtime and number of recursions for NearestPlanes decoding are listed, as well as the resulting success probability and expected total runtime. We also list the corresponding symmetric bit-security levels according to the guidelines of Lenstra and Verheul [LV01]. The unit of all runtimes is seconds, and all times and probabilities are given by their base-two logarithms. For comparison, the last two rows give our security estimates for two parameter sets from [MR09].

n	q	s	Per-User Key ($\mathbf{P}$)	Full Key ($\mathbf{P}$ & $\bar{\mathbf{A}}$)	Ctext ($\mathbf{c}$)	Msg Expan	Security Level
96	2053	7.11	1.4×10^5	2.4×10^5	2.5×10^3	19.3	64 (1992)
128	2053	6.77	1.8×10^5	3.6×10^5	2.8×10^3	22.0	107 (2048)
144	4093	9.35	2.2×10^5	4.7×10^5	3.3×10^3	25.5	140 (2092)
160	4093	9.17	2.5×10^5	5.5×10^5	3.5×10^3	27.0	185 (2150)
192	4093	8.87	2.9×10^5	7.4×10^5	3.8×10^3	30.0	293 (2292)
136	2003	13.01	2.8×10^6	5.8×10^6	2.9×10^3	22.6	191 (2158)
214	16381	7.37	2.4×10^6	6.4×10^6	4.8×10^3	18.7	262 (2251)

Figure 6: Sizes (in bits) of public keys and ciphertexts for the cryptosystem described in Section 3. In each case, the message size is $\ell = 128$ bits. The "message expansion" factor is the ratio of ciphertext size to plaintext size. The symmetric bit-security level is taken from Figure 5.

Figure 5 contains estimates for total expected runtime for representative parameters n, q, and s. In addition, it also shows the corresponding symmetric bit-security level b and "secure until (at least) year Y" estimates, using the heuristic of Lenstra and Verheul [LV01] to allow for comparison with similar works. (As a point of comparison, the heuristic treats DES as secure until 1982, and incorporates advances in computational power due to Moore's Law and increased budgets over time.) Since our experiments ran on a single 2.3GHz processor, the symmetric bit security b and year Y are computed as follows:

$$b = \lg(T) + \lg(2300) - \lg(60 \cdot 60 \cdot 24 \cdot 365.25) - \lg(5 \cdot 10^5) + 56,$$
$$Y = \lfloor (b - 56) \cdot 30/23 + 1982 \rfloor,$$

where T is the total expected runtime on our computer. We also give a table of keysizes for all suggested parameters in Figure 6.

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. 2, 10, 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 and 14.

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

[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, and 9.

[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 7.

[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 pp. 17 and 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, 5, 6, and 17.

[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, and 10.

[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, 4, 5, 6, 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 p. 3.

[Sch03] C.-P. Schnorr. Lattice reduction by random sampling and birthday methods. In *STACS*, pages 145–156. 2003. On p. 12.

[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 p. 11.

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