Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus *

Nicholas Genise
ngenise@eng.ucsd.edu
UCSD

Daniele Micciancio
daniele@cs.ucsd.edu
UCSD

June 2, 2017

Abstract

We present improved algorithms for gaussian preimage sampling using the lattice trapdoors of (Micciancio and Peikert, CRYPTO 2012). The MP12 work only offered a highly optimized algorithm for the on-line stage of the computation in the special case when the lattice modulus q is a power of two. For arbitrary modulus q, the MP12 preimage sampling procedure resorted to general lattice algorithms with complexity cubic in the bitsize of the modulus (or quadratic, but with substantial preprocessing and storage overheads). Our new preimage sampling algorithm (for any modulus q) achieves linear complexity, and has very modest storage requirements. As an additional contribution, we give a simple, new off-line quasi-linear time perturbation sampling algorithm, with performance similar to the expected running time of an efficient method proposed by (Ducas and Nguyen, Asiacrypt 2012) for power-of-two cyclotomics, but derived through drastically different methods. All our algorithms are fairly simple, with small hidden constants, and offer a practical alternative to use the MP12 trapdoor lattices in a broad range of cryptographic applications.

1 Introduction

Lattice cryptography provides powerful techniques to build a wide range of advanced cryptographic primitives, like identity based encryption [27, 19, 9, 3, 1, 2], attribute based encryption [14, 13, 15, 29, 12], some types of fully homomorphic encryption and signatures [11, 10, 28, 31, 21], group signatures [32, 17, 36, 37, 46] and much more (e.g., see [48, 43, 8, 49, 53, 5, 38, 30]). Most of the advanced applications of lattice cryptography rely on a notion of strong lattice trapdoor, introduced in [27], which allows to sample points from an n-dimensional lattice L with a gaussian-like distribution. This gaussian sampling operation is often the main bottleneck in the implementation of advanced cryptographic functions that make use of strong lattice trapdoors, and improving the methods to generate and use lattice trapdoors has been the subject of several investigations [4, 27, 6, 47].

The current state of the art in lattice trapdoor generation and sampling is given by [43], which introduces a new notion of lattice trapdoor, specialized to the type of q-ary lattices used in cryptography, i.e., integer lattices $L \subseteq \mathbb{Z}^n$ that are periodic modulo $q \cdot \mathbb{Z}^n$. Building on techniques from [47], this algorithm includes both an on-line and an off-line stage, and [43] focuses on improving the complexity of the on-line stage, which is far more critical in applications. Unfortunately, the most efficient algorithms proposed in [43] for (the on-line stage of) preimage sampling only apply to lattices with modulus $q = 2^k$ equal to a power of 2 (or, more generally, the power $q = p^k$ of a small prime p,) which is not compatible with the functional or efficiency requirements of many applications. Moreover, only the on-line stage of [43] takes full advantage of the structure of algebraic lattices [42, 40, 41] typically employed in the efficient instantiation of lattice

*Research supported in part by the DARPA SafeWare program. Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA.

	MP12	MP12	**This work**
modulus q	2^k	any	any
G-Sampling precomp.	—	$O(\log^3 q)$	—
G-Sampling space	$O(\log q)$	$O(\log^2 q)$	$O(\log q)$
G-Sampling time	$O(n \log q)$	$O(n \log^2 q)$	$O(n \log q)$

Table 1: Running time and storage of the on-line (G-sampling) algorithm. G-Sampling running times are scaled by a factor n to take into account that each sample requires n independent calls to the underlying G-sampling operation.

cryptography, and essential to reduce the running time of lattice operations from quadratic (in the lattice dimension) to quasi-linear. A straigthforward implementation of the off-line stage (e.g., using a generic Cholesky decomposition algorithm) completely destroys the algebraic structure, and degrades the running time of the (off-line) algorithm from quasi-linear to quadratic or worse. For lattices over "power-of-two" cyclotomic rings (the most popular class of algebraic lattices used in cryptography), a much faster algorithm for the off-line stage is described in [23, Section 6], which uses a combination of lazy floating-point techniques and numerical analysis methods to improve the expected running time of the off-line computation from quadratic to quasilinear, but at the cost of storing the result of a potentially slower than quasi-linear precomputation.[1]

Our Contribution: We present new, improved algorithms for gaussian preimage sampling using the lattice trapdoors of [43]. Specifically, we present a new algorithm (for the on-line stage) capable of handling any modulus q (including the large prime moduli required by some applications) and still achieve the same level of performance of the specialized algorithm of [43] for power-of-two modulus $q = 2^k$. This improves the running time of [43] for arbitrary modulus from cubic $\log^3 q$ (or quadratic $\log^2 q$, using precomputation and a substantial amount of storage) to just linear in $\log q$ and with minimal storage requirements.

As an additional contribution, we present an alternative to the off-line perturbation generation technique of [23] for power-of-two cyclotomic rings, which takes full advantage of the algebraic structure of ring lattices. Inspired by [24] (FFO), the perturbation algorithm presented here owes its efficiency to a structured factorization similar to that of FFO but solves a different problem altogether. Our approach generalizes to other cyclotomic rings, and, used in combination with techniques from [41], achieves similar performance for any cyclotomic of smooth order.

The G-sampling improvements are summarized in Table 1. We believe the methods of derivation in both of the optimal algorithms presented are of deep interest to lattice cryptography. The improvements are not just asymptotic: our new algorithms are fairly simple, with small hidden constants, and include a careful choice of the parameters that allows to implement most steps using only integer arithmetic on very small numbers. The concrete efficiency of our algorithms, in a range of different cryptographic applications, has been recently confirmed by independent implementation efforts [33, 34, 20, 22]. Taken together, our new algorithms provide a very efficient method to instantiate the lattice trapdoor sampling procedures of [43] in the ring setting, without any restriction on the factorization of the modulus q, and offering a ready-to-use solution to cryptographic applications that make use of lattice preimage sampling.

Technical details In order to describe our techniques, we need first to provide more details on the lattice trapdoor sampling problem. Given a lattice L and a target point $\mathbf{t}$, the lattice gaussian sampling problem asks to generate (possibly with the help of some trapdoor information) a random lattice point $\mathbf{v} \in L$ with probability proportional to $\exp(-c\|\mathbf{v} - \mathbf{t}\|^2)$. Building on techniques from [47], this problem is solved in [43] by mapping L to a fixed (key independent) lattice G^n, generating a gaussian sample in G^n, and then

[1] The methods sketched in [23, Section 6] suggest a $\sqrt{n}$ iteration of the Denman-Beavers algorithm on anti-cyclic matrices yielding a $\tilde{O}(n^{1.5})$ precomputation.

mapping the result back to L. (The linear function T mapping G^n to L serves as the trapdoor.) Without further adjustments, this produces a lattice point in L with ellipsoidal gaussian distribution, with covariance which depends on the linear transformation T. In order to produce spherical samples (as required[2] by applications), [43] employs a perturbation technique from [47] which adds some noise (with complementary covariance) to the target $\mathbf{t}$, before using it as a center for the G^n-lattice sampling operation. In summary, the sampling algorithm of [47, 43] consists of two stages:

- an off-line (target independent) stage, which generates perturbation vectors with covariance matrix defined by the trapdoor transformation T, and
- an on-line (target dependent) stage which generates gaussian samples from an (easy to sample) lattice G^n.

Not much attention is paid in [43] to the perturbation generation, as it does not depend on the target vector $\mathbf{t}$, and it is far less time critical in applications.[3] As for the on-line stage, one of the properties that make the lattice G^n easy to sample is that it is the orthogonal sum of n copies of a low dimensional lattice G, of dimension $\log q$. So, even using generic algorithms with quadratic running time, G sampling takes a total of $O(n \log^2 q)$ operations. For moduli $q = n^{O(1)}$ polynomial in the lattice dimension n, this results in quasilinear running time $O(n \log^2 n)$. However, since the G-sampling operation directly affects the on-line running time of the signing algorithm, even a polylogarithmic term $\log^2 q$ can be highly undesirable. To this end, [43] gives a particularly efficient (and easy to implement) algorithm for G-lattice sampling when the lattice modulus $q = 2^k$ is a power of 2 (or more generally, a power $q = p^k$ of a small prime p.) The running time of this specialized G-sampling algorithm is $\log q$, just linear in the lattice dimension, and has minimal (constant) storage requirements. Thanks to its simplicity and efficiency, this algorithm has easily found its way in concrete implementations of lattice based cryptographic primitives (e.g., see [7]), largely solving the problem of efficient lattice sampling for $q = 2^k$. However, setting q to a power of 2 (or more generally, the power of a small prime), may be incompatible with applications and other techniques used in lattice cryptography, like ABE schemes [12] and fast implementation via the number theoretic transform [39, 41]. For arbitrary modulus q, [43] falls back to generic algorithms (for arbitrary lattices) with quadratic complexity. This may still be acceptable when the modulus q is relatively small. But it is nevertheless undesirable, as even polylogarithmic factors have a significant impact on the practical performance of cryptographic functions (easily increasing running times by an order of magnitude), and can make applications completely unusuable when the modulus $q = \exp(n)$ is exponentially large. A concrete example that well illustrates the limitations of [43] is the recent conjuction obfuscator of [16], which requires the modulus q to be prime with bitsize $\log(q) = O(n)$ linear in the security parameter. In this setting, the specialized algorithm of [43] (for $q = 2^k$) is not applicable, and using a generic algorithm slows down the on-line stage by a factor $O(n)$. Another, less drastic, example is the arithmetic circuit ABE scheme of [12] where q is $O(2^{n^\epsilon})$ for some fixed $0 < \epsilon < 1/2$.

Unfortunately, the specialized algorithm from [43] makes critical use of the structure of the G-basis when $q = 2^k$, and is not easily adapted to other moduli. (See Section 3 for details.) In order to solve this problem we resort to the same technique used in [47, 43] to generate samples from arbitrary lattices: we map G to an even simpler lattice D using an easy to compute linear transformation T', perform the gaussian sampling in D, and map the result back to G. As usual, the error shape is corrected by including a perturbation term with appropriate covariance matrix. The main technical problem to be solved is to find a suitable linear transformation T' such that D can be efficiently sampled and perturbation terms can be easily generated. In Section 3 we demonstrate a choice of transformation T' with all these desirable properties. In particular, using a carefully chosen transformation T', we obtain lattices D and perturbation matrices that are triangular, sparse, and whose entries admit a simple (and efficiently computable) closed formula expression. So, there is not even a need to store these sparse matrices explicitly, as their entries can be easily computed on the fly. This results in a G-sampling algorithm with linear running time, and minimal

[2] More generally, applications require samples to be generated according to a distribution that does not depend on the trapdoor/secret key.

[3] E.g., in lattice based digital signature schemes [27, 43], the off-line computation depends only on the secret key, and can be performed in advance without knowing the message to be signed.

(constant) space requirements, beyond the space necessary to store the input, output and randomness of the algorithm.

Next, in Section 4, we turn to the problem of efficiently generating the perturbations of the off-line stage. Notice that generating these perturbations is a much harder problem than the one faced when mapping G to D (via T'). The difference is that while G, D, T' are fixed (sparse, carefully designed) matrices, the transformation T is a randomly chosen matrix that is used as secret key. In this setting, there is no hope to reduce the computation time to linear in the lattice dimension, because even reading/writing the matrix T can in general take quadratic space. Still, when using algebraic lattices, matrix T admits a compact (linear size) representation, and one can reasonably hope for faster perturbation generation algorithms, but [43] gives no indication of how to achieve this. The off-line algorithm of [43] includes both a preprocessing (matrix factorization) and postprocessing (lattice rounding) stage. Implementing the preprocessing using standard Cholesky decomposition techniques (as mentioned in [43] for arbitrary lattices) immediately destroys all the algebraic structure in the trapdoor, and results in very poor performance. An asymptotically superior preprocessing method, which preserves the algebraic structure, is proposed in [23] for the important case of power-of-two cyclotomic and cyclic rings). More specifically, [23, Section 6] observes that when working in such rings, one can replace the Cholesky decomposition with a factorization over ring elements that admits both a compact representation and an efficient algorithm to compute it. We present an alternative algorithm that uses the subring structure of power-of-two cyclotomic rings to completely bypass the need to precompute any matrix factorization, and directly generates perturbation vectors in the target lattice. As mentioned before, at the heart of the algorithm presented here is a matrix factorization inspired by [24].

2 Preliminaries

We denote the complex numbers as $\mathbb{C}$, the real numbers as $\mathbb{R}$, the rational numbers as $\mathbb{Q}$, and the integers as $\mathbb{Z}$. A number is denoted by a lower case letter, $n \in \mathbb{Z}$ for example. We denote the conjugate of a complex number y as y^*. When q is a positive integer, $\log q$ is short for its rounded up logarithm in base two, $\lceil \log_2 q \rceil$. The index set of the first n natural numbers is $[n] = \{1, \ldots, n\}$. Vectors are denoted by bold lower case letters, $\mathbf{v}$, and are in column form ($\mathbf{v}^T$ is a row vector) unless stated otherwise. The inner product of two vectors is $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$. We denote matrices with bold upper case letters $\mathbf{B}$ or with upper case Greek letters (for positive-definite matrices). In addition, we denote the transpose of a matrix as $\mathbf{B}^T$, and its Hermitian transpose as $\mathbf{B}^\dagger$. The entry of $\mathbf{B}$ in row i and column j is denoted $B_{i,j}$. Unless otherwise stated, the norm of a vector is the ℓ_2 norm. The norm of a matrix $\|\mathbf{B}\| = \max_i \|\mathbf{b}_i\|$ is the maximum norm of its column vectors. Given two probability distributions over a countable domain D, the statistical distance between them is $\Delta(X, Y) = \frac{1}{2} \sum_{\omega \in D} |X(\omega) - Y(\omega)|$. In order to avoid tracing irrelevant terms in our statistical distance computations, we define $\hat{\epsilon} = \epsilon + O(\epsilon^2)$. For a random variable X, we denote its expectation as $\mathbb{E}[X]$. We denote a random variable x sampled from a distribution $\mathcal{A}$ as $x \leftarrow \mathcal{A}$. A random variable distributed as $\mathcal{A}$ is denoted $x \sim \mathcal{A}$.

2.1 Linear Algebra

The *Gram-Schmidt orthogonalization* of an ordered set of linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ is $\widetilde{\mathbf{B}} = \{\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_k\}$ where each $\widetilde{\mathbf{b}}_i$ is the component of $\mathbf{b}_i$ orthogonal to $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$. An *anti-cylic* matrix is an $n \times n$ matrix of the form

$$\begin{bmatrix} a_0 & -a_{n-1} & \ldots & -a_1 \\ a_1 & a_0 & \ldots & -a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \ldots & a_0 \end{bmatrix}.$$

For any two (symmetric) matrices $\Sigma, \Gamma \in \mathbb{R}^{n \times n}$, we write $\Sigma \succeq \Gamma$ if $\mathbf{x}^T(\Sigma - \Gamma)\mathbf{x} \geq 0$ for all (nonzero) vectors $\mathbf{x} \in \mathbb{R}^n$, and $\Sigma \succ \Gamma$ if $\mathbf{x}^T(\Sigma - \Gamma)\mathbf{x} > 0$. It is easy to check that $\succeq$ is a partial order relation. Relations

$\preceq$ and $\prec$ are defined symmetrically. When one of the two matrices $\Gamma = s\mathbf{I}$ is scalar, we simply write $\Sigma \succeq s$ or $\Sigma \preceq s$. A symmetric matrix $\Sigma \in \mathbb{R}^{n\times n}$ is called *positive definite* if $\Sigma \succ 0$, and *positive semidefinite* if $\Sigma \succeq 0$. Equivalently, Σ is positive semidefinite if and only if it can be written as $\Sigma = \mathbf{B}\mathbf{B}^T$ for some (square) matrix $\mathbf{B}$, called a *square root* of Σ and denoted $\mathbf{B} = \sqrt{\Sigma}$. (Notice that any $\Sigma \succ 0$ has infinitely many square roots $\mathbf{B} = \sqrt{\Sigma}$.) Σ is positive definite if and only if its square root $\mathbf{B}$ is a square nonsingular matrix. When $\mathbf{B}$ is upper (resp. lower) triangular, the factorization $\Sigma = \mathbf{B}\mathbf{B}^T$ is called the upper (resp. lower) triangular *Cholesky decomposition* of Σ. The Cholesky decomposition of any positive definite $\Sigma \in \mathbb{R}^{n\times n}$ can be computed with $O(n^3)$ floating point arithmetic operations. For any scalar s, $\Sigma \succ s$ if all eigenvalues of Σ are strictly greater than s. In particular, positive definite matrices are nonsingular.

For any $n\times n$ matrix $\mathbf{S}$ and non-empty index sets $I, J \subseteq \{1,\ldots,n\}$, we write $\mathbf{S}[I,J]$ for the $|I|\times|J|$ matrix obtained by selecting the elements at positions $(i,j) \in I\times J$ from $\mathbf{S}$. When $I = J$, we write $\mathbf{S}[I]$ as a shorthand for $\mathbf{S}[I,I]$. Notice that for any nonsingular matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, and index set $I \subseteq \{1,\ldots,n\}$, the submatrix $\mathbf{S}[I]$ is also nonsingular. For any nonsingular matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ and index partition $I \cup \bar{I} = \{1,\ldots,n\}$, $I \cap \bar{I} = \emptyset$, the $I \times I$ matrix

$$\mathbf{S}/I = \mathbf{S}[I] - \mathbf{S}[I,\bar{I}]\cdot\mathbf{S}[\bar{I}]^{-1}\cdot\mathbf{S}[\bar{I},I]$$

is called the *Schur complement* of $\mathbf{S}[\bar{I}]$, often denoted by $\mathbf{S}/\mathbf{S}[\bar{I}] = \mathbf{S}/I$. In particular, if $\mathbf{S} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{D}\end{bmatrix}$ then the Schur complement of $\mathbf{A}$ is the matrix $\mathbf{S}/\mathbf{A} = \mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$. For any index set I, a symmetric matrix $\mathbf{S}$ is positive definite if and only if both $\mathbf{S}[I]$ and its Schur's complement $\mathbf{S}/\mathbf{S}[I]$ are positive definite.

Let $\Sigma = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{D}\end{bmatrix} \succ 0$. We can factor Σ in terms of a principal submatrix, say $\mathbf{D}$, and its Schur complement, $\Sigma/\mathbf{D} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T$, as follows:

$$\Sigma = \begin{bmatrix}\mathbf{I} & \mathbf{B}\mathbf{D}^{-1}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\Sigma/\mathbf{D} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{D}^{-1}\mathbf{B}^T & \mathbf{I}\end{bmatrix}.$$

The next two theorems regarding the spectra of principal submatrices and Schur complements of positive definite matrices are used in Section 4. In both theorems, λ_i is the ith (in non-increasing order, with multiplicity) eigenvalue of a symmetrix matrix.

Theorem 2.1 (Cauchy) *For any symmetric matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$, $I \subseteq \{1,\ldots,n\}$ and $1 \le i \le |I|$*

$$\lambda_i(\mathbf{S}) \ge \lambda_i(\mathbf{S}[I]) \ge \lambda_{i+n-|I|}(\mathbf{S}).$$

Theorem 2.2 ([54, Corollary 2.3]) *For any positive definite $\Sigma \in \mathbb{R}^{n\times n}$, $I \subseteq \{1,\ldots,n\}$ and $1 \le i \le |I|$*

$$\lambda_i(\Sigma) \ge \lambda_i(\Sigma/I) \ge \lambda_{i+n-|I|}(\Sigma).$$

In other words, the eigenvalues of principal submatrices and Schur complements of a positive definite matrix are bounded from below and above by the smallest and largest eigenvalues of the original matrix, respectively.

2.2 Gaussians and Lattices

A *lattice* $\Lambda \subset \mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^n$. Specifically, a lattice of *rank* k is the integer span $\mathcal{L}(\mathbf{B}) = \{z_1\mathbf{b}_1 + \cdots + z_k\mathbf{b}_k \mid z_i \in \mathbb{Z}\}$ of a basis $\mathbf{B} = \{\mathbf{b}_1,\ldots,\mathbf{b}_k\} \subset \mathbb{R}^n$ ($k \le n$). There are infinitely many bases for a given lattice since right multiplying a basis by a unimodular transformation gives another basis. The *dual lattice* of Λ, denoted by Λ^*, is the lattice $\{\mathbf{x} \in \mathrm{span}(\Lambda) \mid \langle\mathbf{x},\Lambda\rangle \subseteq \mathbb{Z}\}$. It is easy to see that $\mathbf{B}^{-T}$ is a basis for $\mathcal{L}(\mathbf{B})^*$ for a full rank lattice ($n = k$).

The n-dimensional *gaussian* function $\rho : \mathbb{R}^n \to (0,1]$ is defined as $\rho(\mathbf{x}) := \exp(-\pi\|\mathbf{x}\|^2)$. Applying an invertible linear transformation $\mathbf{B}$ to the gaussian function yields

$$\rho_{\mathbf{B}}(\mathbf{x}) = \rho(\mathbf{B}^{-1}\mathbf{x}) = \exp(-\pi\cdot\mathbf{x}^T\Sigma^{-1}\mathbf{x})$$

with $\Sigma = \mathbf{B}\mathbf{B}^T \succ 0$. For any $\mathbf{c} \in \text{span}(\mathbf{B}) = \text{span}(\Sigma)$, we also define the shifted gaussian function (centered at $\mathbf{c}$) as $\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{x}) = \rho_{\sqrt{\Sigma}}(\mathbf{x} - \mathbf{c})$. Normalizing the function $\rho_{\mathbf{B},\mathbf{c}}(\mathbf{x})$ by the measure of $\rho_{\mathbf{B},\mathbf{c}}$ over the span of $\mathbf{B}$ gives the *continuous gaussian distribution* with covariance $\Sigma/(2\pi)$, denoted by $D_{\sqrt{\Sigma},\mathbf{c}}$. Let $S \subset \mathbb{R}^n$ be any discrete set in $\mathbb{R}^n$, then $\rho_{\sqrt{\Sigma}}(S) = \sum_{\mathbf{s}\in S} \rho_{\sqrt{\Sigma}}(\mathbf{s})$. The *discrete gaussian* distribution over a lattice Λ, denoted by $D_{\Lambda,\sqrt{\Sigma},\mathbf{c}}$, is defined by restricting the support of the distribution to Λ. Specifically, a sample $\mathbf{y} \leftarrow D_{\Lambda,\sqrt{\Sigma},\mathbf{c}}$ has probability mass function $\rho_{\sqrt{\Sigma},\mathbf{c}}(\mathbf{x})/\rho_{\sqrt{\Sigma},\mathbf{c}}(\Lambda)$ for all $\mathbf{x} \in \Lambda$. Discrete gaussians on lattice cosets $\Lambda + \mathbf{c}$, for $\mathbf{c} \in \text{span}(\Lambda)$, are defined similarly setting $\Pr\{\mathbf{y} \leftarrow D_{\Lambda+\mathbf{c},\sqrt{\Sigma},\mathbf{p}}\} = \rho_{\sqrt{\Sigma},\mathbf{p}}(\mathbf{y})/\rho_{\sqrt{\Sigma},\mathbf{p}}(\Lambda + \mathbf{c})$ for all $\mathbf{y} \in \Lambda + \mathbf{c}$. For brevity we let $D_{\Lambda+\mathbf{c},\sqrt{\Sigma},\mathbf{p}}(\mathbf{y}) := \Pr\{\mathbf{y} \leftarrow D_{\Lambda+\mathbf{c},\sqrt{\Sigma},\mathbf{p}}\}$.

For a lattice Λ and any (typically small) positive $\epsilon > 0$, the *smoothing parameter* $\eta_\epsilon(\Lambda)$ [44] is the smallest $s > 0$ such that $\rho(s \cdot \Lambda^*) \leq 1 + \epsilon$. More generally, for any positive definite matrix Σ and lattice $\Lambda \subset \text{span}(\Sigma)$, we write $\sqrt{\Sigma} \geq \eta_\epsilon(\Lambda)$, or $\Sigma \succeq \eta_\epsilon^2(\Lambda)$, if $\rho(\sqrt{\Sigma}^T \cdot \Lambda^*) \leq 1 + \epsilon$. The reader is referred to [44, 27, 47] for additional information on the smoothing parameter. Here we recall two bounds and a discrete gaussian convolution theorem to be used later.

Lemma 2.1 ([27, Lemma 3.1]) *Let $\Lambda \subset \mathbb{R}^n$ be a lattice with basis $\mathbf{B}$, and let $\epsilon > 0$. Then,*

$$\eta_\epsilon(\Lambda) \leq \|\widetilde{\mathbf{B}}\| \sqrt{\log(2n(1 + 1/\epsilon))/\pi}.$$

Lemma 2.2 ([47, Lemma 2.5]) *For any full rank n-dimensional lattice Λ, vector $\mathbf{c} \in \mathbb{R}^n$, real $\epsilon \in (0, 1)$, and positive definite $\Sigma \succeq \eta_\epsilon^2(\Lambda)$,*

$$\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{c}) \in \left[\frac{1-\epsilon}{1+\epsilon}, 1\right] \cdot \rho_{\sqrt{\Sigma}}(\Lambda).$$

Theorem 2.3 ([47, Theorem 3.1]) *For any vectors $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{R}^n$, lattices $\Lambda_1, \Lambda_2 \subset \mathbb{R}^n$, and positive definite matrices $\Sigma_1, \Sigma_2 \succ 0$, $\Sigma = \Sigma_1 + \Sigma_2 \succ 0$, $\Sigma_3^{-1} = \Sigma_1^{-1} + \Sigma_2^{-1} \succ 0$, if $\sqrt{\Sigma_1} \succeq \eta_\epsilon(\Lambda_1)$ and $\sqrt{\Sigma_3} \succeq \eta_\epsilon(\Lambda_2)$ for some $0 < \epsilon \leq 1/2$, then the distribution*

$$X = \{\mathbf{x} \mid \mathbf{p} \leftarrow D_{\Lambda_2+\mathbf{c}_2,\sqrt{\Sigma_2}}, \mathbf{x} \leftarrow D_{\Lambda_1+\mathbf{c}_1,\sqrt{\Sigma_1},\mathbf{p}}\}$$

is within statistical distance $\Delta(X, Y) \leq 8\epsilon$ from the discrete gaussian $Y = D_{\Lambda_1+\mathbf{c}_1,\sqrt{\Sigma}}$.

Below, we have the correctness theorem for the standard, randomized version of Babai's nearest plane algorithm. The term *statistically close* is the standard cryptographic notion of negligible statistical distance. Precisely, a function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if for every $c > 1$ there exists an N such that for all $n > N$, $f(n) < n^{-c}$. We emphasize that the algorithm reduces to sampling $D_{\mathbb{Z},s,c}$.

Theorem 2.4 ([27, Theorem 4.1]) *Given a full-rank lattice basis $\mathbf{B} \in \mathbb{R}^{n\times n}$, a parameter $s \geq \|\tilde{\mathbf{B}}\|\omega(\sqrt{\log n})$, and a center $\mathbf{c} \in \mathbb{R}^n$, there is an $O(n^2)$-time, with a $O(n^3)$-time preprocessing, probabilistic algorithm whose output is statistically close to $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$.*

2.3 Cyclotomic Fields

Let n be a positive integer. The *n-th cyclotomic field over* $\mathbb{Q}$ is the number field $\mathcal{K}_n = \mathbb{Q}[x]/(\Phi_n(x)) \cong \mathbb{Q}(\zeta)$ where ζ is an *n-th primitive root of unity* and $\Phi_n(x)$ is the minimal polynomial of ζ over $\mathbb{Q}$. The nth *cyclotomic ring* is $\mathcal{O}_n = \mathbb{Z}[x]/(\Phi_n(x))$. Let $\varphi(n)$ be Euler's totient function. $\mathcal{K}_n$ is a $\varphi(n)$-dimensional $\mathbb{Q}$-vector space, and we can view $\mathcal{K}_n$ as a subset of $\mathbb{C}$ by viewing ζ as a complex primitive n-th root of unity.

Multiplication by a fixed element f, $g \mapsto f \cdot g$, is a linear transformation on $\mathcal{K}_n$ as a $\mathbb{Q}$-vector space. We will often view field elements as $\varphi(n)$-dimensional rational vectors via the *coefficient embedding*. This is defined by $f(x) = \sum_{i=0}^{\varphi(n)-1} f_i x^i \mapsto (f_0, \cdots, f_{\varphi(n)-1})^T$ mapping a field element to its vector of coefficients under the *power basis* $\{1, x, \cdots, x^{\varphi(n)-1}\}$ (or equivalently $\{1, \zeta, \cdots, \zeta^{\varphi(n)-1}\}$). We can represent a field element as the matrix in $\mathbb{Q}^{\varphi(n)\times\varphi(n)}$ that represents the linear transformation by its multiplication in the coefficient embedding. This matrix is called a field element's coefficient *multiplication* matrix. When n is a power of two, an element's coefficient multiplication matrix is anti-cyclic.

SAMPLEG$(q = b^k, s, u)$
  for $i = 0, \ldots, k-1$:
    $x_i \leftarrow \mathcal{D}_{b\mathbb{Z}+u,s}$
    $u := (u - x_i)/b \in \mathbb{Z}$.
  return $(x_0, \ldots, x_{k-1})$.

Figure 1: A sampling algorithm for G-lattices when the modulus q is a perfect power of the base b. The algorithm is implicitly parametrized by a base b and dimension k.

An *isomorphism* from the field F to the field K is a bijection $\theta : F \to K$ such that $\theta(fg) = \theta(f)\theta(g)$, and $\theta(f + g) = \theta(f) + \theta(g)$ for all $f, g \in F$. An *automorphism* is an isomorphism from a field to itself. For example, if we view the cyclotomic field $\mathcal{K}_n$ as a subset of the complex numbers, then the *conjugation* map $f(\zeta) \mapsto f(\zeta)^* = f(\zeta^*)$ is an automorphism and can be computed in linear time $O(n)$. In power-of-two cyclotomic fields, the conjugation of a field element corresponds to the matrix transpose of an element's anti-cyclic multiplication matrix.

Another embedding is the *canonical* embedding which maps an element $f \in \mathcal{K}_n$ to the vector of evaluations of f, as a polynomial, at each root of $\Phi_n(x)$. When n is a power of two, the linear transformation between the coefficient embedding and the canonical embedding is a scaled isometry.

Let n be a power of two, then the field $\mathcal{K}_{2n}$ is a two-dimensional $\mathcal{K}_n$-vector space as see by splitting a polynomial $f(x) \in \mathcal{K}_{2n}$ into $f(x) = f_0(x^2) + x \cdot f_1(x^2)$ for $f_i \in \mathcal{K}_n$. Now, we can view the linear transformation given by multiplication by a f as a linear transformation over $\mathcal{K}_n \times \mathcal{K}_n \cong \mathcal{K}_{2n}$. Let $\phi_{2n} : \mathcal{K}_{2n} \to \mathbb{Q}^{n \times n}$ be the injective ring homomorphism from the field to an element's anti-cyclic matrix. Then, we have the following relationship where $\mathbf{P}$ below is a simple re-indexing matrix known as a stride permutation (increasing evens followed by increasing odds in $\{0, 1, \ldots, n-1\}$),

$$\mathbf{P}\phi_n(f)\mathbf{P}^T = \begin{bmatrix} \phi_{n/2}(f_0) & \phi_{n/2}(x \cdot f_1) \\ \phi_{n/2}(f_1) & \phi_{n/2}(f_0) \end{bmatrix}.$$

3 Sampling G-lattices

For any positive integers $b \geq 2$, $k \geq 1$ and non-negative interger $u < b^k$, we write $[u]_b^k$ for the base-b expansion of u, i.e., the unique vector $(u_0, \ldots, u_{k-1})$ with entries $0 \leq u_i < b$ such that $u = \sum_i u_i b^i$. Typically, $b = 2$ and $[u]_2^k$ is just the k-digits binary representation of u, but larger values of b may be used to obtain interesting efficiency trade-offs. Throughout this section, we consider the values of b and k as fixed, and all definitions and algorithms are implicitly parametrized by them.

In this section we study the so-called G-lattice sampling problem, i.e., the problem of sampling the discrete Gaussian distribution on a lattice coset

$$\Lambda_u^\perp(\mathbf{g}^T) = \{\mathbf{z} \in \mathbb{Z}^k : \mathbf{g}^T\mathbf{z} = u \bmod q\}$$

where $q \leq b^k$, $u \in \mathbb{Z}_q$ and $\mathbf{g} = (1, b, \ldots, b^{k-1})$. A very efficient algorithm to solve this problem is given in [43] for the special case when $q = b^k$ is a power of the base b. The algorithm, shown in Figure 1, is very simple. This algorithm reduces the problem of sampling the k-dimensional lattice coset $\Lambda_u^\perp(\mathbf{g}^T)$ for $u \in \mathbb{Z}_q$ to the much simpler problem of sampling the *one-dimensional* lattice cosets $u + b\mathbb{Z}$ for $u \in \mathbb{Z}_b$. The simplicity of the algorithm is due to the fact that, when $q = b^k$ is an exact power of b, the lattice $\Lambda^\perp(\mathbf{g}^T)$ has a very special basis

$$\mathbf{B}_{b^k} = \begin{bmatrix} b & & & & \\ -1 & b & & & \\ & -1 & \ddots & & \\ & & \ddots & b & \\ & & & -1 & b \end{bmatrix}$$

which is sparse, triangular, and with small integer entries. (In particular, its Gram-Schmidt orthogonalization $\widetilde{\mathbf{B}}_{b^k} = b\mathbf{I}$ is a scalar matrix.) As a result, the general lattice sampling algorithm of [35, 27] (which typically requires $O(k^3)$-time preprocessing, and $O(k^2)$ storage and online running time) can be specialized to the much simpler algorithm in Figure 1 that runs in linear time $O(k)$, with minimal memory requirements and no preprocessing at all.

We give a specialized algorithm to solve the same sampling problem when $q < b^k$ is an arbitrary modulus. This is needed in many cryptographic applications where the modulus q is typically a prime. As already observed in [43] the lattice $\Lambda^{\perp}(\mathbf{g}^T)$ still has a fairly simple and sparse basis matrix

$$\mathbf{B}_q = \begin{bmatrix} b & & & & q_0 \\ -1 & b & & & q_1 \\ & -1 & \ddots & & \vdots \\ & & \ddots & b & q_{k-2} \\ & & & -1 & q_{k-1} \end{bmatrix}$$

where $(q_0, \ldots, q_{k-1}) = [q]_b^k = \mathbf{q}$ is the base-b representation of the modulus q. This basis still has good geometric properties, as all vectors in its (left-to-right) Gram-Schmidt orthogonalization have length at most $O(b)$. So, it can be used with the algorithm of [35, 27] to generate good-quality gaussian samples on the lattice cosets with small standard deviation. However, since the basis is no longer triangular, its Gram-Schmidt orthogonalization is not sparse anymore, and the algorithm of [35, 27] can no longer be optimized to run in linear time as in Figure 1. In applications where $q = n^{O(1)}$ is polynomial in the security parameter n, the matrix dimension $k = O(\log n)$ is relatively small, and the general sampling algorithm (with $O(k^2)$ storage and running time) can still be used with an acceptable (albeit significant) performance degradation. However, for larger q this becomes prohibitive in practice. Moreover, even for small q, it would be nice to have an optimal sampling algorithm with $O(k)$ running time, linear in the matrix dimension, as for the exact power case. Here we give such an algorithm, based on the convolution methods of [47], but specialized with a number of concrete technical choices that result in a simple and very fast implementation, comparable to the specialized algorithm of [43] for the exact power case.

The intuitive reader may notice that the alternating columns of $\mathbf{B}_q$, $\mathbf{b}_1, \mathbf{b}_3, \ldots$ and $\mathbf{b}_2, \mathbf{b}_4, \ldots$, are pairwise orthogonal. Let us call these sets $\mathbf{B}_1$ and $\mathbf{B}_2$, respectively. Then, another basis for $\Lambda^{\perp}(\mathbf{g}^T)$ is $(\mathbf{B}_1, \mathbf{B}_2, \mathbf{q})$ and this might suggest that the GSO of this basis is sparse. Unfortunately, this leads to a GSO of $(\mathbf{B}_1, \mathbf{B}_2^*, \mathbf{q}^*)$ where $\mathbf{B}_2^*$ is a dense, upper triangular block. Let $\mathbf{b}$ be the $i-th$ vector in $\mathbf{B}_2$. Then, there are $2 + i - 1$ non-orthogonal vectors to $\mathbf{b}$ preceding it in $\mathbf{B}_1$ and $\mathbf{B}_2^*$, filling in the upper portion of $\mathbf{b}$.

Regarding the MP12 sampler as a whole, another intuitive attempt would be to use Peikert's randomized rounder without perturbation [47] and simply sample $\Lambda^{\perp}(\mathbf{g}^T)$ with a skewed distribution (where we would simply change Σ_p in Section 4 to account for this). The goal is a linear algorithm since $\mathbf{B}_q$ is sparse. This intuition is correct since we can use back substitution twice from the decomposition of $\mathbf{B}_q$ into the product of two triangular matrices given in the next subsection. This decomposition of $\mathbf{B}_q$ is needed for this method to yield a linear time algorithm.

Overview The idea is the following. Instead of sampling $\Lambda_u^{\perp}(\mathbf{g}^T)$ directly, we express the lattice basis $\mathbf{B}_q = \mathbf{TD}$ as the image (under a linear transformation $\mathbf{T}$) of some other matrix $\mathbf{D}$ with very simple (sparse, triangular) structure. Next, we sample the discrete gaussian distribution (say, with variance σ^2) on an appropriate coset of $\mathcal{L}(\mathbf{D})$. Finally, we map the result back to the original lattice applying the linear transformation $\mathbf{T}$ to it. Notice that, even if $\mathcal{L}(\mathbf{D})$ is sampled according to a spherical gaussian distribution, the resulting distribution is no longer spherical. Rather, it follows an ellipsoidal gaussian distribution with (scaled) covariance $\sigma^2\mathbf{TT}^T$. This problem is solved using the convolution method of [47], i.e., initially adding a perturbation with complementary covariance $s^2\mathbf{I} - \sigma^2\mathbf{TT}^T$ to the target, so that the final output has covariance $\sigma^2\mathbf{TT}^T + (s^2\mathbf{I} - \sigma^2\mathbf{TT}^T) = s^2\mathbf{I}$. In summary, at a very high level, the algorithm performs (at least implicitly) the following steps:

1. Compute the covariance matrix $\Sigma_1 = \mathbf{T}\mathbf{T}^T$ and an upper bound r on the spectral norm of $\mathbf{T}\mathbf{T}^T$
2. Compute the complementary covariance matrix $\Sigma_2 = r^2\mathbf{I} - \Sigma_1$
3. Sample $\mathbf{p} \leftarrow D_{\Lambda_1,\sigma\sqrt{\Sigma_2}}$, from some convenient lattice Λ_1 using the Cholesky decomposition of Σ_2
4. Compute the preimage $\mathbf{c} = \mathbf{T}^{-1}(\mathbf{u} - \mathbf{p})$
5. Sample $\mathbf{z} \leftarrow D_{\mathcal{L}(\mathbf{D}),-\mathbf{c},\sigma}$
6. Output $\mathbf{u} + \mathbf{T}\mathbf{z}$

The technical challenge is to find appropriate matrices $\mathbf{T}$ and $\mathbf{D}$ that lead to a very efficient implementation of all the steps. In particular, we would like $\mathbf{T}$ to be a very simple matrix (say, sparse, triangular, and with small integer entries) so that $\mathbf{T}$ has small spectral norm, and both linear transformations $\mathbf{T}$ and $\mathbf{T}^{-1}$ can be computed efficiently. The matrix $\mathbf{D}$ (which is uniquely determined by $\mathbf{B}$ and $\mathbf{T}$) should also be sparse and triangular, so that the discrete gaussian distribution on the cosets of $\mathcal{L}(\mathbf{D})$ can be efficiently sampled. Finally (and this is the trickiest part in obtaining an efficient instantiation) the complementary covariance matrix $\Sigma_2 = r^2\mathbf{I} - \Sigma_1$ should also have a simple Cholesky decoposition $\Sigma_2 = \mathbf{L}\mathbf{L}^T$ where $\mathbf{L}$ is triangular, sparse and with small entries, so that perturbations can be generated efficiently. Ideally, all matrices should also have a simple, regular structure, so that they do not need to be stored explicitly, and can be computed on the fly with minimal overhead.

In the next subsection we provide an instantiation that satisfies all of these properties. Next, in Subsection 3.2 we describe the specialized sampling algorithm resulting from the instantiation, and analyze its correctness and efficiency properties.

3.1 Instantiation

In this subsection, we describe a specific choice of linear transformations and matrix decompositions that satisfies all our desiderata, and results in a very efficient instantiation of the convolution sampling algorithm on G-lattices.

A tempting idea may be to map the lattice basis $\mathbf{B}_q$ to the basis $\mathbf{B}_{b^k}$, and then use the efficient sampling algorithm from Figure 1. However, this does not quite work because it results in a pretty bad transformation $\mathbf{T}$ which has both poor geometrical properties and a dense matrix representation. It turns out that a very good choice for a linear transformation $\mathbf{T}$ is given precisely by the matrix $\mathbf{T} = \mathbf{B}_{b^k}$ describing the basis when q is a power of b. We remark that $\mathbf{T}$ is used as a linear transformation, rather than a lattice basis. So, the fact that it equals $\mathbf{B}_{b^k}$ does not seem to carry any special geometric meaning, it just works! In particular, what we do here should not be confused with mapping $\mathbf{B}_q$ to $\mathbf{B}_{b^k}$. The resulting factorization is

$$\mathbf{B}_q = \begin{bmatrix} b & & & & q_0 \\ -1 & b & & & q_1 \\ & -1 & \ddots & & \vdots \\ & & \ddots & b & q_{k-2} \\ & & & -1 & q_{k-1} \end{bmatrix} = \begin{bmatrix} b & & & & \\ -1 & b & & & \\ & -1 & \ddots & & \\ & & \ddots & b & \\ & & & -1 & b \end{bmatrix} \begin{bmatrix} 1 & & & & d_0 \\ & 1 & & & d_1 \\ & & \ddots & & \vdots \\ & & & 1 & d_{k-2} \\ & & & & d_{k-1} \end{bmatrix} = \mathbf{B}_{b^k}\mathbf{D}$$

where the entries of the last column of $\mathbf{D}$ are defined by the recurrence $d_i = \frac{d_{i-1}+q_i}{b}$ with initial condition $d_{-1} = 0$. Notice that all the d_i are in the range $[0, 1)$, and $b^{i+1} \cdot d_i$ is always an integer. In some sense, sampling from $\mathcal{L}(\mathbf{D})$ is even easier than sampling from $\mathcal{L}(\mathbf{B}_{b^k})$ because the first $k-1$ columns of $\mathbf{D}$ are orthogonal and the corresponding coordinates can be sampled independently in parallel. (This should be contrasted with the sequential algorithm in Figure 1.)

We now look at the geometry and algorithmic complexity of generating perturbations. The covariance matrix of $\mathbf{T} = \mathbf{B}_{b^k}$ is given by

$$\Sigma_1 = \mathbf{B}_{b^k}\mathbf{B}_{b^k}^T = \begin{bmatrix} b^2 & -b & & & & \\ -b & (b^2+1) & -b & & & \\ & \ddots & \ddots & \ddots & & \\ & & -b & (b^2+1) & -b & \\ & & & -b & (b^2+1) & \end{bmatrix}.$$

The next step is to find an upper bound r^2 on the spectral norm of Σ_2, and compute the Cholesky decomposition $\mathbf{L}\mathbf{L}^T$ of the complementary covariance matrix $\Sigma_2 = r^2\mathbf{I} - \Sigma_1$. By the Gershgorin circle theorem, all eigenvalues of Σ_1 are in the range $(b \pm 1)^2$. So, we may set $r = b + 1$. Numerical computations also suggest that this choice of r is optimal, in the sense that the spectral norm of Σ_1 approaches $b + 1$ as k tends to infinity. The Cholesky decomposition is customarily defined by taking $\mathbf{L}$ to be a *lower* triangular matrix. However, for sampling purposes, an upper triangular $\mathbf{L}$ works just as well. It turns out that using an upper triangular $\mathbf{L}$ in the decomposition process leads to a much simpler solution, where all (squared) entries have a simple, closed form expression, and can be easily computed on-line without requiring any preprocessing computation or storage. (By contrast, numerical computations suggest that the standard Cholesky decomposition with lower triangular $\mathbf{L}$ is far less regular, and even precomputing it requires exponentially higher precision arithmetic than our upper triangular solution.) So, we let $\mathbf{L}$ be an upper triangular matrix, and set $r = b + 1$.

For any r, the perturbation's covariance matrix $\Sigma_2 = r^2\mathbf{I} - \Sigma_1$ has Cholesky decomposition $\Sigma_2 = \mathbf{L} \cdot \mathbf{L}^T$ where $\mathbf{L}$ is the sparse upper triangular matrix defined by the following equations:

$$\mathbf{L} = \begin{bmatrix} l_0 & h_1 & & & \\ & l_1 & h_2 & & \\ & & \ddots & \ddots & \\ & & & & h_{k-1} \\ & & & & l_{k-1} \end{bmatrix} \quad \text{where} \quad \begin{array}{rcll} l_0^2 + h_1^2 &=& r^2 - b^2 & \\ l_i^2 + h_{i+1}^2 &=& r^2 - (b^2+1) & (i = 1, \ldots, k-2) \\ l_{k-1}^2 &=& r^2 - (b^2+1) & \\ l_i h_i &=& b & (i = 1, \ldots, k-1) \end{array}$$

It can be easily verified that these equations have the following simple closed form solution:

$$r = b + 1, \quad l_0^2 = b\left(1 + \frac{1}{k}\right) + 1, \quad l_i^2 = b\left(1 + \frac{1}{k-i}\right), \quad h_{i+1}^2 = b\left(1 - \frac{1}{k-i}\right) \tag{1}$$

We observe that also the inverse transformation $\mathbf{B}_{b^k}^{-1}$ has a simple, closed-form solution: the ith column of $\mathbf{B}_{b^k}^{-1}$ equals $(0, \cdots, 0, \frac{1}{b}, \ldots, (\frac{1}{b})^{k-i})$. Notice that this matrix is not sparse, as it has $O(k^2)$ nonzero entries. However, there is no need to store it and the associated transformation can still be computed in linear time by solving the sparse triangular system $\mathbf{T}\mathbf{x} = \mathbf{b}$ by back-substitution.

3.2 The Algorithm

The sampling algorithm, SampleG, is shown in Figure 2. It takes as input a modulus q, an integer variance s, a coset u of $\Lambda^\perp(\mathbf{g}^T)$, and outputs a sample statistically close to $D_{\Lambda_u^\perp(\mathbf{g}^T),s}$. SampleG relies on subroutines Perturb and SampleD where Perturb(σ) returns a perturbation, $\mathbf{p}$, statistically close to $D_{\mathcal{L}(\Sigma_2),\sigma\cdot\sqrt{\Sigma_2}}$, and SampleD($\sigma, \mathbf{c}$) returns a sample $\mathbf{z}$ such that $\mathbf{D}\mathbf{z}$ is statistically close to $D_{\mathcal{L}(\mathbf{D}),-\mathbf{c},\sigma}$. Both Perturb and SampleD are instantiations of the randomized nearest plane algorithm [35, 27]. In addition, Perturb and SampleD rely on a subroutine SampleZ(σ, t) which returns a sample statistically close to $D_{\mathbb{Z},t,\sigma}$.

Assuming constant time sampling for SampleZ and scalar arithmetic, SampleG runs in time $O(k)$. The scalars c_i in SampleG, representing $\mathbf{c} = \mathbf{B}_{b^k}^{-1}(\mathbf{u} - \mathbf{p})$, and d_i in SampleD, representing the last column of $\mathbf{D}$, are rational numbers of the form x/b^i for a small integer x and $i \in [k]$. The numbers l_i, h_i are positive numbers less than $\sqrt{2b+1}$. Recent results suggest [51, 45], longer than double floating point precision is

SampleG$(s, \mathbf{u} = [u]_b^k, \mathbf{q} = [q]_b^k)$
  $\sigma := s/(b+1)$
  $\mathbf{p} \leftarrow$ Perturb(σ)
  for $i = 0, \ldots, k-1$:
    $c_i := (c_{i-1} + u_i - p_i)/b$
  $\mathbf{z} \leftarrow$ SampleD$(\sigma, \mathbf{c})$
  for $i = 0, \ldots, k-2$:
    $t_i := b \cdot z_i - z_{i-1} + q_i \cdot z_{k-1} + u_i$
  $t_{k-1} := q_{k-1} \cdot z_{k-1} - z_{k-2} + u_{k-1}$
  return t

Perturb(σ)
  $\beta := 0$
  for $i = 0, \ldots, k-1$:
    $c_i := \beta/l_i$, and $\sigma_i := \sigma/l_i$
    $z_i \leftarrow$ SampleZ(σ_i, c_i)
    $\beta := -z_i h_i$
  $p_0 := (2b+1)z_0 + bz_1$
  for $i := 1, \ldots, k-1$:
    $p_i := b(z_{i-1} + 2z_i + z_{i+1})$
  return p

SampleD$(\sigma, \mathbf{c})$
  $z_{k-1} \leftarrow$ SampleZ$(\sigma/d_{k-1}, -c_{k-1}/d_{k-1})$
  $\mathbf{c} := \mathbf{c} - z_{k-1}\mathbf{d}$
  for $i \in \{0, \ldots, k-2\}$:
    $z_i \leftarrow$ SampleZ$(\sigma, -c_i)$
  return z

Figure 2: Sampling algorithm for G-lattices for any modulus $q < b^k$. The algorithms take b and k as implicit parameters, and SampleG outputs a sample with distribution statistically close to $D_{\Lambda_u^\perp(\mathbf{g}^T),s}$. Any scalar with an index out of range is 0, i.e. $c_{-1} = z_{-1} = z_k = 0$.

not needed for l_i and h_i for commonly used parameters. However, one could instantiate SampleZ with the lazy floating point techniques of [23] and SampleG would be quasi-linear time in k instead of linear in k, if higher precision is needed.

By combining the bounds in Theorems 2.3 and 2.4, the variance s can be practically as small as in the case when $q = p^k$ for a small prime p. The algorithms store floating point numbers c_i, d_i, h_i, and l_i for a total storage of $O(k)$ floating point numbers, but can be adapted to constant time storage since they are determined by simple recurrence relations (c_i, d_i) or simple formulas (h_i, l_i). We conclude with the statement of correctness in the form of the following corollary. [4]

Corollary 3.1 *Let* $0 < \epsilon \leq 1/2$ *be such that* $\frac{s^2}{(b+1)^2} \geq \eta_\epsilon(\mathcal{L}(\mathbf{D}))$ *and* $\sqrt{\Sigma_3} \geq \eta_\epsilon(\mathcal{L}([b+1]^2\mathbf{B}_{b^k}^{-1} - \mathbf{B}_{b^k}^T))$ *where* $\Sigma_3^{-1} = \frac{(b+1)^2}{s^2}\mathbf{I} + [s^2\mathbf{B}_{b^k}^{-1}\mathbf{B}_{b^k}^{-t} - \frac{s^2}{(b+1)^2}\mathbf{I}]^{-1}$. *In addition, let* $\frac{s}{b+1} \geq \max\{\|\widetilde{\mathbf{D}}\|, \|\widetilde{\mathbf{L}}^T\|\} \cdot \omega(\sqrt{\log n})$. *Then,* SampleG *returns a perturbation within a statistical distance* $O(\epsilon)$ *from* $D_{\Lambda_u^\perp(\mathbf{g}^T),s}$ *for any* $q < b^k$.

4 Perturbation Sampling in Cyclotomic Rings

The lattice preimage sampling algorithm of [43] requires the generation of $n(2 + \log q)$-dimensional gaussian perturbation vectors $\mathbf{p}$ with covariance $\Sigma_p = s^2 \cdot \mathbf{I} - \alpha^2\mathbf{T} \cdot \mathbf{T}^T$ where $\mathbf{T} \in \mathbb{Z}^{(2+\log q)n \times n \log q}$ is a matrix with small entries serving as a lattice trapdoor, α is a small constant factor and s is an upper bound on the spectral norm of $\alpha\mathbf{T}$. In [43] this is accomplished using the Cholesky factorization of Σ_p, which takes $O(n \log q)^3$ precomputation and $O(n \log q)^2$ storage and running time. (All integer arithmetics can be performed modulo q, and time and space complexity bounds are in terms of arithmetic operations in $\mathbb{Z}_q$.)

The trapdoor matrix $\mathbf{T}$ of [43] has some additional structure: $\mathbf{T}^T = [\bar{\mathbf{T}}^T, \mathbf{I}]$ for some $\bar{\mathbf{T}} \in \mathbb{Z}^{2n \times n \log q}$. Moreover, when working with algebraic lattices, $\bar{\mathbf{T}} = \phi_n(\tilde{\mathbf{T}})$ is the image (under a ring embedding $\phi_n \colon R_n \to \mathbb{Z}^{n \times n}$) of some matrix $\tilde{\mathbf{T}} \in R_n^{2 \times \log q}$ with entries in a ring R_n of rank n. (Most commonly, $R_n = \mathcal{O}_{2n} = \mathbb{Z}[x]/(x^n + 1)$ is the ring of integers of the $(2n)$th cyclotomic field $\mathcal{K}_{2n}$ for $n = 2^k$ a power of two.) In [7] it

[4] The keen reader will notice that there is another error term, $\epsilon'(n)$, negligible in n in the statistical distance bound in Corollary 3.1 from Theorem 2.4. We assume $\epsilon \geq \epsilon'$ to get our $O(\epsilon)$ term.

is observed that, using the sparsity of Σ_p, the preprocessing storage and on-line computation cost of noise perturbation reduce to $O(n^2 \log q)$.[5] This is a factor $\log q$ improvement over a generic implementation, but it is still quadratic in the main security parameter n. This can be a significant improvement in practice, but the overall cost of the algorithm remains substantial. When using generic trapdoors $\bar{\mathbf{T}} \in \mathbb{Z}^{2n \times n \log q}$, there is little hope to improve the running time below $O(n^2 \log q)$, because just reading the matrix $\bar{\mathbf{T}}$ takes this much time. However, when using algebraic lattices, the trapdoor $\bar{\mathbf{T}} = \phi_n(\tilde{\mathbf{T}})$ admits a compact representation $\tilde{\mathbf{T}}$ consisting of only $2n \log q$ integers, so one may hope to reduce the running time to linear or quasi-linear in n. Unfortunately, all efficiency advantages of working with structured lattices are lost when computing the Cholesky decomposition of Σ_p as suggested in [43], because the Cholesky decomposition does not respect the ring structure and produces arbitrary matrices over $\mathbb{R}$.

A method for discrete gaussian sampling with covariance described by a ring element in R_n in quasi-linear time is given in [23]. A similar quasi-linear sampler can also be obtained by using the CRT transformation (see [41]) to diagonalize the matrix described by the ring element. Both methods require the randomized rounding sampler of [47]. The recent algorithm of [24] (FFO) shows how to use the subring structure of cyclic rings in order to perform an efficient version of Babai's nearest plane algorithm on cyclic lattices (or the randomized version of Klein). Though similar in approach, it seems unclear how to adapt FFO to sampling $\mathbb{Z}^{n(2+\log q)}$ with a structured, anti-cyclic covariance. The natural embedding from a cyclotomic ring to a cyclic ring mentioned in [24] necessarily maps a positive definite anti-cyclic matrix to a positive semi-definite one with trivial (0) eigenvalues. This is seen through the Chinese Remainder Theorem and its idempotents.

In this section we give an alternative, more direct algorithm to generate integer perturbation vectors $\mathbf{p}$ with covariance Σ_p when $\bar{\mathbf{T}} = \phi_n(\tilde{\mathbf{T}})$. Our algorithm takes full advantage of the ring structure of R_n, compactly representing Σ_p and all other matrices generated during the execution of the algorithm as the image of matrices with entries in the ring R_n. In particular, similarly to [23], our algorithm has time and space complexity quasi-linear in n, but it does not require any preprocessing/storage. The algorithm can be expressed in a modular way as the combination of three steps:

1. First, the problem of sampling a $O(n \log q)$-dimensional integer vectors $\mathbf{p}$ with covariance Σ_p is reduced to the problem of sampling a $2n$-dimensional integer vector with covariance expressed by a 2×2 matrix over R_n.

2. Next, the problem of sampling with covariance in $R_n^{2\times 2}$ is reduced to sampling two n-dimensional vectors, each with a covariance expressed by a single ring element in R_n.

3. Finally, if $n > 1$, the sampling problem with covariance in R_n is reduced to sampling an n-dimensional perturbation with covariance expressed by a 2×2 matrix over the smaller ring $R_{n/2}$.

Iterating the last two steps $\log n$ times reduces the original problem to sampling in $R_1 = \mathbb{Z}$. Details about each step are given in the next subsections. We remark that the algorithm is described as a recursive procedure only for simplicity of presentation and analysis, and it can be implemented just as easily using a simple nested loop, similarly to many FFT-like algorithms. We first describe an algorithm to generate perturbations from continuous gaussians, which admits a simpler proof, but requires the use of high precision floating point arithmetics. Then, we describe an integer-based version of the algorithm which directly generates perturbations from discrete gaussians over $\mathbb{Z}^{n(2+\log q)}$. The continuous gaussian perturbation generation method presented here is given primarily as a warm up to the main algorithm to generate discrete gaussian perturbations.

4.1 Algorithm for Floating Point Perturbations

The sampling algorithm for continuous gaussians is shown in Figure 3. The entry point of the algorithm is the SamplePz procedure, which takes as input two integer parameters n, q, a matrix $\tilde{\mathbf{T}} \in R_n^{2\times \log q}$ and

[5] Sparsity also reduces the preprocessing running time to $O(\log q \cdot n^2 + n^3) = O(n^3)$, but still cubic in n.

SAMPLEP$(n, q, s^2, \alpha^2, \tilde{\mathbf{T}})$
$z = (\alpha^{-2} - s^{-2})^{-1}$
$a = s^2 - z \cdot \tilde{\mathbf{T}}_0 \tilde{\mathbf{T}}_0^T$
$b = -z \cdot \tilde{\mathbf{T}}_0 \tilde{\mathbf{T}}_1^T$
$d = s^2 - z \cdot \tilde{\mathbf{T}}_1 \tilde{\mathbf{T}}_1^T$
for $i = 0, \ldots, (n \log q - 1)$:
$\mathbf{q}_i \leftarrow$ SAMPLER$(s^2 - \alpha^2)$
$\mathbf{p} \leftarrow$SAMPLE2(a, b, d)
return $\left(\mathbf{p} - \left(\frac{\alpha}{s^2 - \alpha^2}\right)\mathbf{T}\mathbf{q}, \mathbf{q}\right)$

SAMPLE2(a, b, d)
$p_1 \leftarrow$ SAMPLEF(d)
$p_0 \leftarrow$ SAMPLEF$(a - bd^{-1}b^*)$
return $(p_0 + (bd^{-1})p_1, p_1)$

SAMPLEF(f)
if $\dim(f) = 1$ **return** SAMPLER(f)
else let $f(x) = f_0(x^2) + x \cdot f_1(x^2)$
$(p_0, p_1) \leftarrow$ SAMPLE2(f_0, f_1, f_0)
let $p(x) = p_0(x^2) + x \cdot p_1(x^2)$
return p

Figure 3: Sampling algorithm SAMPLEP for floating point perturbations where $\mathbf{T} = \phi_n(\tilde{\mathbf{T}})$ is a compact trapdoor over a power of two cyclotomic ring. Note, $\tilde{\mathbf{T}}_i$ is a *row* vector over R_n for each $i \in \{0, 1\}$. The algorithm uses a subroutine SAMPLER(σ^2) to sample a 1-dimensional gaussian with variance σ^2 centered at 0.

two positive real numbers s^2, α^2, and is expected to produce an $n(2 + \log q)$-dimensional vector $\mathbf{p}$ with (non-spherical) gaussian distribution $\mathcal{D}_{\sqrt{\Sigma_p}}$ of covariance

$$\Sigma_p = s^2 \cdot \mathbf{I} - \alpha^2 \begin{bmatrix} \phi_n(\tilde{\mathbf{T}}) \\ \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \phi_n(\tilde{\mathbf{T}})^T & \mathbf{I} \end{bmatrix} = \begin{bmatrix} s^2\mathbf{I} - \alpha^2 \phi_n(\tilde{\mathbf{T}}) \cdot \phi_n(\tilde{\mathbf{T}})^T & -\alpha^2 \phi_n(\tilde{\mathbf{T}}) \\ -\alpha^2 \phi_n(\tilde{\mathbf{T}})^T & (s^2 - \alpha^2)\mathbf{I} \end{bmatrix} \tag{2}$$

The algorithm calls two subroutines:

- SAMPLER$(s^2 - \alpha^2)$ which samples a one-dimensional gaussian variable of variance $s^2 - \alpha^2$, and can be implemented using any standard technique, and
- SAMPLE2(a, b, d), which, on input three ring elements a, b, d compactly describing a positive definite matrix
$$\Sigma_2 = \begin{bmatrix} \phi_n(a) & \phi_n(b) \\ \phi_n(b)^T & \phi_n(d) \end{bmatrix},$$
is expected to sample a $(2n)$-dimensional vector $p \leftarrow D_{\sqrt{\Sigma_2}}$ with covariance Σ_2.

In turn, SAMPLE2 (also described in Figure 3) makes use of a procedureSAMPLEF(f) which on input a ring element f with positive definite $\phi_n(f)$, returns a sample $p \leftarrow D_{\sqrt{\phi_n(f)}}$.

Correctness The proof of correctness of the algorithm is based on the following lemma.

Lemma 4.1 *For any positive definite matrix* $\Sigma = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix}$*, the process*

1. $\mathbf{p}_1 \leftarrow D_{\sqrt{\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T}}$
2. $\mathbf{p}_2 \leftarrow D_{\sqrt{\mathbf{D}}}$
3. *return* $\begin{bmatrix} \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{I} \end{bmatrix} \mathbf{p}_2 + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{p}_1$

produces a vector with gaussian distribution $D_{\sqrt{\Sigma}}$.

Proof: Let $\mathbf{p}_2, \mathbf{p}_1$ be independent $\mathbf{0}$-mean multivariate gaussian samples with respective covariances $\mathbf{D}$ and $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T$. Due to independence, the output $(\mathbf{B}\mathbf{D}^{-1}\mathbf{p}_2 + \mathbf{p}_1, \mathbf{p}_2)$ is gaussian with covariance

$$\begin{bmatrix} \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{I} \end{bmatrix} \mathbf{D} \begin{bmatrix} \mathbf{D}^{-T}\mathbf{B}^T & \mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T) \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix}$$

as claimed. □

Here we discuss the linear algebra behind the algorithm. As mentioned in section 2, the anti-cyclic matrix of a field element can be represented as a two-by-two block matrix with multiplication matrices in the smaller field, $\phi_{n/2}(\cdot)$, as blocks by permuting the row and column indices according to the stride permutation as in the FFT (permuting the increasing elements of $\{0, 1, \ldots, n-1\}$ to the increasing evens followed by the increasing odds). One more step yields a useful factorization after the permutation

$$\phi_n(f) = \begin{bmatrix} \phi_{n/2}(f_0) & \phi_{n/2}(x \cdot f_1) \\ \phi_{n/2}(f_1) & \phi_{n/2}(f_0) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{1} & \phi_{n/2}(f_1 f_0^{-1}) \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \phi_{n/2}(f_0 - f_1 f_0^{-1} f_1^*) & \mathbf{0} \\ \mathbf{0} & \phi_{n/2}(f_0) \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \phi_{n/2}(f_0^{-1} f_1^*) & \mathbf{1} \end{bmatrix}.$$

Now, we have a diagonalization of $\phi_n(f)$, with entries over smaller and smaller rings, by repeating the above factorization on $\phi_{n/2}(f_0 - f_1 f_0^{-1} f_1^*)$ and $\phi_{n/2}(f_0)$ (including the re-indexing matrices). This decomposition is similar to the FFO decomposition [24], except here we add a step isolating the Schur complement-principal submatrix pair $\phi_{n/2}(f_0 - f_1 f_0^{-1} f_1^*), \phi_{n/2}(f_0)$. Notice, $\phi_{n/2}(x \cdot f_1) = \phi_{n/2}(f_1)^T = \phi_{n/2}(f_1^*)$ when $\phi_n \succ 0$ as is the case for the input f. As a result, SampleF can simply perform the convolution in Lemma 4.1 with a call to Sample2 when the dimension is greater than one and sample $D_{\sqrt{f}}$ over $\mathbb{R}$ otherwise.

We prove the correctness of the sampling algorithm in two steps, corresponding to the three procedures SampleP, Sample2 and SampleF. Starting from the base case of SampleF, the correctness of SampleF implies the correctness of Sample2, which in turn implies the correctness of SampleP.

Corollary 4.1 *If* SampleF *correctly samples the distribution* $D_{\sqrt{\phi_n(f)}}$*, then* Sample2 *correctly samples the distribution* $D_{\sqrt{\Sigma_2}}$*.*

Proof: The corollary follows by instantiating Lemma 4.1 with $\mathbf{A} = \phi_n(a)$, $\mathbf{B} = \phi_n(b)$ and $\mathbf{D} = \phi_n(d)$. □

Corollary 4.2 *If* Sample2 *correctly samples the distribution* $D_{\sqrt{\Sigma_2}}$*, then* SampleP *correctly samples the distribution* $D_{\sqrt{\Sigma_p}}$*.*

Proof: The corollary follows by instantiating Lemma 4.1 with $\Sigma = \Sigma_p$, equivalently
$\mathbf{A} = s^2\mathbf{I} - \alpha^2 \phi_n(\tilde{\mathbf{T}}) \cdot \phi_n(\tilde{\mathbf{T}})^T$, $\mathbf{B} = -\alpha^2 \phi_n(\tilde{\mathbf{T}})$ and $\mathbf{D} = (s^2 - \alpha^2)\mathbf{I}$. □

Efficiency Multiplications are done in the field $\mathcal{K}_i$, for an element's dimension $i \in \{1, 2, \ldots, 2n\}$, in time $\tilde{O}(i)$ by using the Chinese remainder transform (CRT) [41].

By treating scalar arithmetic as constant time, SampleP has a time complexity of $\tilde{O}(n \log q)$ because the transformation by $\mathbf{T}$ is $O(n \log n \log q)$ and SampleF has complexity $O(n \log^2 n)$ (represented by the recurrence $R(n) = 2R(n/2) + (n/2) \log n/2$). The algorithm requires $n \log q$ scalar storage for the trapdoor $\tilde{\mathbf{T}}$.

$\textsc{SamplePz}(n, q, s, \alpha, \tilde{\mathbf{T}})$
  $z = (\alpha^{-2} - s^{-2})^{-1}$
  $a := s^2 - z\tilde{\mathbf{T}}_0\tilde{\mathbf{T}}_0^T$
  $b := -z\tilde{\mathbf{T}}_0\tilde{\mathbf{T}}_1^T$
  $d := s^2 - z\tilde{\mathbf{T}}_1\tilde{\mathbf{T}}_1^T$

  for $i = 0, \ldots, (n \log q - 1)$:
    $q_i \leftarrow \textsc{SampleZ}(s^2 - \alpha^2)$
  $\mathbf{c} := -\frac{\alpha^2}{s^2 - \alpha^2}\tilde{\mathbf{T}}\mathbf{q}$
  $\mathbf{p} \leftarrow \textsc{Sample2z}(a, b, d, \mathbf{c},)$
  return $(\mathbf{p}, \mathbf{q})$

$\textsc{Sample2z}(a, b, d, \mathbf{c})$
  let $\mathbf{c} = (c_0, c_1)$
  $q_1 \leftarrow \textsc{SampleFz}(d, c_1)$
  $c_0 := c_0 + bd^{-1}(q_1 - c_1)$
  $q_0 \leftarrow \textsc{SampleFz}(a - bd^{-1}b^*, c_0)$
  return (q_0, q_1)

$\textsc{SampleFz}(f, \mathbf{c})$
  if $\dim(f) = 1$ **return** $\textsc{SampleZ}(f, c)$
  else let $f(x) = f_0(x^2) + x \cdot f_1(x^2)$
    $\mathbf{c}' := P_{stride(n)}(\mathbf{c})$
    $(q_0, q_1) \leftarrow \textsc{Sample2z}(f_0, f_1, f_0, \mathbf{c}')$
    let $q(x) = q_0(x^2) + x \cdot q_1(x^2)$
    return q

Figure 4: Sampling algorithm SamplePz for integer perturbations where $\mathbf{T} = \phi_n(\tilde{\mathbf{T}})$ is a compact trapdoor over a power of two cyclotomic ring. Note, $\tilde{\mathbf{T}}_i$ is a *row* vector over R_n for each $i \in \{0, 1\}$. The algorithm uses a subroutine $\textsc{SampleZ}(\sigma^2, t)$ which samples a discrete gaussian over $\mathbb{Z}$ with variance σ^2 centered at t. The stride permutation, $P_{stride(n)}$, sends a vector of coefficients to the increasing even indices followed by the increasing odd indices.

4.2 Integer Perturbation Algorithm for Power of Two Cyclotomics

In this subsection we present a discrete version of the SampleP algorithm which produces $n(2 + \log q)$-dimensional perturbations from a *discrete* gaussian in time $\tilde{O}(n \log q)$. The main algorithm SamplePz (and auxiliary algorithms Sample2z, SampleFz) are described in Figure 4, and closely follow their continuous counterparts from Figure 3. We refer the reader to Subsection 4.1 for a discussion of the intuition behind the algorithms, and move straight to their formal analysis.

Correctness One would use the discrete gaussian convolution theorem, Theorem 2.3, in an initial attempt to prove the correctness of the algorithms in Figure 4. However, this would only ensure the correctness of the marginal distributions of $\mathbf{p}$ in SamplePz and q_0 in Sample2z and not their respective joint distributions, $(\mathbf{p}, \mathbf{q})$ and (q_0, q_1). Even if it were enough, tracking the Σ_3 condition in Theorem 2.3 through the recursive calls of the algorithms above is tedious. Instead, we derive a convolution lemma without a Σ_3 condition for the joint distribution of our discrete gaussian convolutions on the simple lattice $\mathbb{Z}^n$.

First, we show the gaussian function $\rho_{\sqrt{\Sigma}}(\cdot)$ factors in a useful manner with respect to a Schur complement decomposition.

Lemma 4.2 *Let* $\Sigma = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix} \succ \mathbf{0}$ *be a positive definite with* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *and* $\mathbf{D} \in \mathbb{R}^{m \times m}$ *and* $\Sigma/\mathbf{D} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T$ *is* $\mathbf{D}$*'s Schur complement, and let* $\mathbf{x}_1 \in \mathbb{R}^n$ *and* $\mathbf{x}_2 \in \mathbb{R}^m$ *be arbitrary. Then, the gaussian function* $\rho_{\sqrt{\Sigma}}(\mathbf{x})$ *factors as* $\rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbf{x}_1 - \mathbf{B}\mathbf{D}^{-1}\mathbf{x}_2) \cdot \rho_{\mathbf{D}}(\mathbf{x}_2) = \rho_{\sqrt{\Sigma}}(\mathbf{x})$ *where* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}^{n+m}$.

Proof:(Sketch) This is seen through defining the inverse of Σ in terms of $\Sigma/\mathbf{D}$ and writing out $\rho_{\sqrt{\Sigma}}(\mathbf{x})$ in terms of $\Sigma/\mathbf{D}$. The matrix factorization

$$\Sigma = \begin{bmatrix} \mathbf{I} & \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Sigma/\mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{D}^{-1}\mathbf{B}^T & \mathbf{I} \end{bmatrix}$$

yields the formula for Σ^{-1} needed to show the result. □

Note, a simple consequence of the above lemma is that the gaussian sum $\rho_{\sqrt{\Sigma}}(\mathbb{Z}^{n+m})$ expands in terms of the gaussian functions $\rho_{\sqrt{\mathbf{D}}}(\cdot)$ and $\rho_{\sqrt{\Sigma/\mathbf{D}}}(\cdot)$,

$$\rho_{\sqrt{\Sigma}}(\mathbb{Z}^{n+m}) = \sum_{\mathbf{y}_2 \in \mathbb{Z}^m} \rho_{\sqrt{\mathbf{D}}}(\mathbf{y}_2) \cdot \rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n - \mathbf{B}\mathbf{D}^{-1}\mathbf{y}_2).$$

We will use the following lemma for the correctness proof. It states that if a discrete gaussian on the integer lattice is wide enough in its slimmest direction, then the lower dimensional discrete gaussians with covariance shaped with principal submatrices of the original are wide enough on their respective $\mathbb{Z}^{n'}$s.

Lemma 4.3 *Let, $\epsilon > 0$, $\Sigma \succ 0$ be a positive definite matrix in $\mathbb{R}^{n\times n}$, and let $I_0 \subset [n]$ be an arbitrary, non-empty subset. If $\Sigma \succeq \eta_\epsilon^2(\mathbb{Z}^n)$, then $\Sigma[I_0] \succeq \eta_\epsilon^2(\mathbb{Z}^{|I_0|})$ and $\Sigma/\bar{I}_0 \succeq \eta_\epsilon^2(\mathbb{Z}^{n-|I_0|})$ for any principal submatrix - Schur complement pair, $(\Sigma[I_0], \Sigma/\bar{I}_0)$, of Σ.*

Proof: Note, a simple consequence of $\Sigma \succeq \eta_\epsilon^2(\mathbb{Z}^n)$ is that Σ's minimum eigenvalue, $\lambda_{min}(\Sigma)$, is greater than $\eta_\epsilon^2(\mathbb{Z}^n)$. Let $\mathbf{M} := \Sigma[I_0] \in \mathbb{R}^{n_0\times n_0}$ for $n_0 = |I_0|$. $\mathbf{M}$ is diagonalizable so let $\mathbf{M} = \mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}$ be its diagonalization. Notice, we have the following inequality from the interlacing theorems which imply $\lambda_{min}(\mathbf{D}) \geq \lambda_{min}(\Sigma)$,

$$\mathbf{x}^T\mathbf{M}\mathbf{x} = \mathbf{x}^T\mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}\mathbf{x} = \mathbf{y}^T\mathbf{\Lambda}\mathbf{y} = \sum_{i\in[n_0]} \lambda_i y_i^2 \geq \lambda_{min}(\Sigma)\|\mathbf{y}\|^2 = \lambda_{min}(\Sigma)\|\mathbf{x}\|^2.$$

Next, we can bound the quantity $\rho_{\mathbf{M}^{-1}}(\mathbb{Z}^{(n_0)*}) = \rho_{\mathbf{M}^{-1}}(\mathbb{Z}^{n_0})$ by $1+\epsilon$:

$$\rho_{\mathbf{M}^{-1}}(\mathbb{Z}^{n_0}) = \sum_{\mathbf{x}\in\mathbb{Z}^{n_0}} e^{-\pi\mathbf{x}^T\mathbf{M}\mathbf{x}} \leq \sum_{\mathbf{x}\in\mathbb{Z}^{n_0}} e^{-\pi\lambda_{min}(\Sigma)\|\mathbf{x}\|^2}$$

$$\leq \sum_{\mathbf{x}\in\mathbb{Z}^{n}} e^{-\pi\lambda_{min}(\Sigma)\|\mathbf{x}\|^2} \leq 1+\epsilon.$$

The jump from $\mathbb{Z}^{n_0}$ to $\mathbb{Z}^n$ comes from the relation $\mathbb{Z}^{n_0} \subset \mathbb{Z}^n$. The proof for the Schur complement is identical. □

Next, we state the analogous lemma to Lemma 4.1.

Lemma 4.4 *For any real $0 < \epsilon \leq 1/2$, positive integers n, m, vector $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{R}^{n+m}$, and positive definite matrix $\Sigma = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{D}\end{bmatrix} \succeq \eta_\epsilon^2(\mathbb{Z}^{n+m})$, $\mathbf{A} \in \mathbb{Z}^{n\times n}$, $\mathbf{B} \in \mathbb{Z}^{n\times m}$, and $\mathbf{D} \in \mathbb{Z}^{m\times m}$ (where $\Sigma/\mathbf{D} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^T$ is the Schur complement of $\mathbf{D}$) the random process*

- $\mathbf{x}_2 \leftarrow D_{\mathbb{Z}^m, \sqrt{\mathbf{D}}, \mathbf{c}_2}$.
- $\mathbf{x}_1 \leftarrow D_{\mathbb{Z}^n, \sqrt{\Sigma/\mathbf{D}}, \mathbf{c}_1 + \mathbf{B}\mathbf{D}^{-1}(\mathbf{x}_2 - \mathbf{c}_2)}$.

produces a vector $(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{Z}^{n+m}$ distributed within statistical distance 2ϵ from $D_{\mathbb{Z}^{n+m}, \sqrt{\Sigma}, \mathbf{c}}$.

Proof:
First, we write out the probability and use Lemma 4.2 to simplify the numerator. Let $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2)$ below.

$$\Pr[\mathbf{x}_1 = \bar{\mathbf{x}}_1, \mathbf{x}_2 = \bar{\mathbf{x}}_2] = \frac{\rho_{\sqrt{\Sigma/\mathbf{D}}}(\bar{\mathbf{x}}_1 - \mathbf{c}_1 - \mathbf{B}\mathbf{D}^{-1}(\bar{\mathbf{x}}_2 - \mathbf{c}_2)) \cdot \rho_{\sqrt{\mathbf{D}}}(\bar{\mathbf{x}}_2 - \mathbf{c}_2)}{\rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n - \mathbf{c}_1 - \mathbf{B}\mathbf{D}^{-1}(\bar{\mathbf{x}}_2 - \mathbf{c}_2)) \cdot \rho_{\sqrt{\mathbf{D}}}(\mathbb{Z}^m - \mathbf{c}_2)}$$

$$= \frac{\rho_{\sqrt{\Sigma}}(\bar{\mathbf{x}} - \mathbf{c})}{\rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n - \mathbf{c}_1 - \mathbf{B}\mathbf{D}^{-1}(\bar{\mathbf{x}}_2 - \mathbf{c}_2)) \cdot \rho_{\sqrt{\mathbf{D}}}(\mathbb{Z}^m - \mathbf{c}_2)}$$

Regarding the denominator, we use Lemma 4.3 to see that $\mathbf{\Sigma}/\mathbf{D} \succeq \eta_\epsilon^2(\mathbb{Z}^n)$ since $\Sigma \succeq \eta_\epsilon^2(\mathbb{Z}^{n+m})$. Now, we can use Lemma 2.2 for the first gaussian sum (dependent on $\bar{\mathbf{x}}_2$) in the denominator to see,

$$\Pr[\mathbf{x}_1 = \bar{\mathbf{x}}_1, \mathbf{x}_2 = \bar{\mathbf{x}}_2] \in \alpha \cdot D_{\mathbb{Z}^{n+m},\sqrt{\Sigma},\mathbf{c}}(\bar{\mathbf{x}}) \cdot \left[\left(\frac{1-\epsilon}{1+\epsilon}\right), 1\right]^{-1}$$

where $\alpha = \frac{\rho_{\sqrt{\Sigma}}(\mathbb{Z}^{n+m}-\mathbf{c})}{\rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n)\cdot\rho_{\sqrt{\mathbf{D}}}(\mathbb{Z}^m-\mathbf{c}_2)}$.

Next, we show $\alpha \approx 1$. Using Lemma 4.2 we expand

$$\rho_{\sqrt{\Sigma}}(\mathbb{Z}^{n+m} - \mathbf{c}) = \sum_{\mathbf{y}_2 \in \mathbb{Z}^m} \rho_{\sqrt{\mathbf{D}}}(\mathbf{y}_2 - \mathbf{c}_2) \cdot \rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n - \mathbf{c}_1 - \mathbf{B}\mathbf{D}^{-1}(\mathbf{y}_2 - \mathbf{c}_2)).$$

The sum $\rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n - \mathbf{c}_1 - \mathbf{B}\mathbf{D}^{-1}(\mathbf{y}_2 - \mathbf{c}_2))$ is approximately $\rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n)$ because $\Sigma/\mathbf{D} \succeq \eta_\epsilon^2(\mathbb{Z}^n)$ as a consequence of Lemma 4.3 and $\Sigma \succeq \eta_\epsilon^2(\mathbb{Z}^{n+m})$. In other words,

$$\rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n - \mathbf{c}_1 - \mathbf{B}\mathbf{D}^{-1}(\mathbf{y}_2 - \mathbf{c}_2)) \in \left[\frac{1-\epsilon}{1+\epsilon}, 1\right] \cdot \rho_{\sqrt{\Sigma/\mathbf{D}}}(\mathbb{Z}^n)$$

and $\alpha \in \left[\left(\frac{1-\epsilon}{1+\epsilon}\right), 1\right]$.

Finally, we have the approximation

$$\Pr[\mathbf{x}_1 = \bar{\mathbf{x}}_1, \mathbf{x}_2 = \bar{\mathbf{x}}_2] \in \left[\left(\frac{1-\epsilon}{1+\epsilon}\right), \left(\frac{1+\epsilon}{1-\epsilon}\right)\right] \cdot D_{\mathbb{Z}^{n+m},\sqrt{\Sigma},\mathbf{c}}(\bar{\mathbf{x}}).$$

Given the restriction on $\epsilon \in (0, 1/2]$, we have the relation we desire

$$\Pr[\mathbf{x}_1 = \bar{\mathbf{x}}_1, \mathbf{x}_2 = \bar{\mathbf{x}}_2] \in [1-4\epsilon, 1+4\epsilon] \cdot D_{\mathbb{Z}^{n+m},\sqrt{\Sigma},\mathbf{c}}(\bar{\mathbf{x}}).$$

□

We now show the correctness of SampleFz implies the correctness of Sample2z, and the correctness of Sample2z implies the correctness of SamplePz.

Corollary 4.3 *If* SampleFz *correctly samples the distribution* $D_{\mathbb{Z}^{n'},\sqrt{\phi_{n'}(f)},\mathbf{c}'}$ *for* $n' \in 1, 2, \ldots, n$ *and* $\Sigma_2 \succeq \eta_\epsilon^2(\mathbb{Z}^{2n'})$*, then* Sample2$_Z$ *samples from a distribution within a statistical distance of* 2ϵ *of* $D_{\mathbb{Z}^{2n'},\sqrt{\Sigma}_2,\mathbf{c}}$ *for any real* $0 < \epsilon \leq 1/2$.

Proof: The corollary follows by instantiating Lemma 4.4 with $\mathbf{A} = \phi_n(a)$, $\mathbf{B} = \phi_n(b)$ and $\mathbf{D} = \phi_n(d)$. □

Corollary 4.3 shows the correctness of SampleFz as well since we can follow the recursions to the base case, the one-dimensional sampler, which we assume is error-less.

Corollary 4.4 *If* Sample2$_Z$ *correctly samples the distribution* $D_{\mathbb{Z}^{2n'},\sqrt{\Sigma}_2},\mathbf{c}$ *and* $\Sigma_p \succeq \eta_\epsilon^2(\mathbb{Z}^{n(2+\log q)})$*, then* SamplePz *returns a sample within a statistical distance of* 2ϵ *of the distribution* $D_{\mathbb{Z}^{n(2+\log q)},\sqrt{\Sigma}_p}$ *for any real* $0 < \epsilon \leq 1/2$.

Proof: The corollary follows by instantiating Lemma 4.4 with $\Sigma := \Sigma_p$, equivalently
$\mathbf{A} = s^2\mathbf{I} - \alpha^2\phi_n(\tilde{\mathbf{T}}) \cdot \phi_n(\tilde{\mathbf{T}})^T$, $\mathbf{B} = -\alpha^2\phi_n(\tilde{\mathbf{T}})$ and $\mathbf{D} = (s^2 - \alpha^2)\mathbf{I}$. □

Finally, we bound the statistical distance between the output of SamplePz and the desired distribution. We need to ensure each discrete gaussian convolution in the algorithm is non-degenerate. The fact that $\Sigma_p \succ 0$ is enough for the continuous algorithms to return a non-degenerate sample, though not for the discrete case. As shown in Lemma 4.4, we need $\Sigma/\mathbf{D} \succeq \eta_\epsilon^2(\mathbb{Z}^{n_0})$ and $\mathbf{D} \succeq \eta_\epsilon^2(\mathbb{Z}^{n_1})$ at *each* of the n discrete gaussian convolutions. Thankfully, this is met through a simple condition on Σ_p as hinted to in Lemma 4.3.

Theorem 4.1 *Let $0 < \epsilon \leq 1/2$. If $\Sigma_p \succeq \eta_\epsilon^2(\mathbb{Z}^{n(2+\log q)})$, then* SamplePz *returns a perturbation within a statistical distance of $O(n \cdot \hat{\epsilon})$ from $D_{\mathbb{Z}^{n(2+\log q)}, \sqrt{\Sigma_p}}$.*

Proof: Since *each* covariance given to SampleFz is a Schur complement or a principal submatrix of a Schur complement of Σ_p, Lemma 4.3 and the interlacing theorems (Theorem 2.1 and Theorem2.2) imply the conditions for Corollaries 4.3 and 4.4 are met. Since there are $\Theta(n)$ convolutions, the output of SamplePz is in the interval

$$[(1-4\epsilon)^{\Theta(n)}, (1+4\epsilon)^{\Theta(n)}] \cdot D_{\mathbb{Z}^{n(2+\log q)}, \sqrt{\Sigma_p}}$$

and the statistical distance is within $O(n\hat{\epsilon})$ by the binomial theorem and ignoring non-linear factors in ϵ. □

A linear degradation in noise quality is the price for the repeated convolutions. This is a minor loss for practical parameters like $\epsilon \approx 2^{-80}$ and n in the high hundreds.

Storage and Efficiency Recent results suggest double precision floating point numbers are enough to preserve security in lattice-based cryptosystems for commonly used parameters [45, 51], but one can use the lazy floating point techniques of [23] for SampleZ and still yield a version of SamplePz that has quasi-linear time complexity on average if longer floating point precision is needed. This would involve tweaking SampleFz and Sample2z to record their path through the tree of recursions and pass the path to SampleZ. Then, SampleZ could re-compute its center and variance in high precision with access to previous samples and the trapdoor $\tilde{\mathbf{T}}$.

SampleFz runs in time $\tilde{O}(n)$, like the continuous case in the previous subsection, and SamplePz runs in time $\tilde{O}(n \log q)$ as a result. Storage consists of storing the trapdoor $\tilde{\mathbf{T}}$ which consists of $2n \log q$ small integers ($O(\log n)$ bits each), and the algorithm stores $4n$ floating point numbers for the input of Sample2z.

4.3 General Cyclotomic Rings

Here we sketch sampling methods for arbitrary cyclotomics. Both the techniques in the previous subsection and the techniques of [23, 47] apply to this setting as well. Let $n = \varphi(n')$, q be a positive integer, $\mathcal{O}_{n'} = \mathbb{Z}[x]/(\Phi_{n'}(x))$ be the n'-th cyclotomic ring with $\mathcal{K}_{n'}$ as the n'-th cyclotomic field over $\mathbb{Q}$, and let $\tilde{\mathbf{T}} \in \mathcal{O}_{n'}^{2\times \log q}$ be a ring trapdoor matrix. Let $\mathrm{rad}(n')$ be the product of all distinct prime divisors of n'. Define the diagonal matrix of a field element, $f \in \mathcal{K}_{n'}$, as $\mathbf{\Psi}(f)_{i,i} = f(\zeta^i)$ where ζ is a complex primitive n'-th root of unity and each i is a distinct element in the group of units modulo n', $i \in \mathbb{Z}_{n'}^*$. This is the multiplication matrix of an element in the canonical embedding. Notice, $\mathbf{\Psi}(f)^\dagger = \mathbf{\Psi}(f^*)$ since conjugation is an automorphism of all cyclotomic fields over $\mathbb{Q}$. We apply $\mathbf{\Psi}(\cdot)$ element-wise to vectors and matrices over $\mathcal{K}_{n'}$.

Now, our goal is to efficiently sample the lattice $D_{\mathcal{O}_{n'}^{2+\log q}, \sqrt{\Sigma_p}}$ where

$$\Sigma_p = s^2\mathbf{I} - \alpha^2 \begin{bmatrix} \mathbf{\Psi}(\tilde{\mathbf{T}}) \\ I \end{bmatrix} \begin{bmatrix} \mathbf{\Psi}(\tilde{\mathbf{T}})^\dagger & I \end{bmatrix}$$

and $\mathbf{\Psi}(\tilde{\mathbf{T}})^\dagger$ is the Hermitian transpose of $\mathbf{\Psi}(\tilde{\mathbf{T}})$. Sampling $D_{\mathcal{O}_{n'}^{2+\log q}, \sqrt{\Sigma_p}}$ reduces to sampling discrete gaussians over $\mathbb{Z}$ in nearly the same steps as the previous subsection with $z = (\alpha^{-2} - s^{-2})^{-1}$:

1. Sampling $\mathbf{p} \leftarrow D_{\mathcal{O}_{n'}^{2+\log q}, \sqrt{\Sigma_p}}$ reduces to sampling $D_{\mathcal{O}_{n'}^2, \sqrt{\Sigma_{2\times 2}}}$ where

$$\Sigma_{2\times 2} = s^2\mathbf{I} - z \cdot \mathbf{\Psi}(\tilde{\mathbf{T}})\mathbf{\Psi}(\tilde{\mathbf{T}})^\dagger = \begin{bmatrix} \mathbf{\Psi}(a) & \mathbf{\Psi}(b) \\ \mathbf{\Psi}(b^*) & \mathbf{\Psi}(d) \end{bmatrix}$$

   by first sampling $\mathbf{p}_2 \leftarrow D_{\mathcal{O}_{n'}^{\log q}, \sqrt{s^2-\alpha^2}}$, updating the randomized center $\mathbf{c} := \frac{-\alpha^2}{s^2-\alpha^2}\mathbf{\Psi}(\tilde{\mathbf{T}})\mathbf{p}_2$, then sampling $\mathbf{p}_1 \leftarrow D_{\mathcal{O}_{n'}^2, \mathbf{c}, \sqrt{\Sigma_{2\times 2}}}$.

2. Sampling $D_{\mathcal{O}_{n'}^2,\mathbf{c},\sqrt{\Sigma_{2\times 2}}}$ reduces to sampling $D_{\mathcal{O}_{n'},\sqrt{\mathbf{\Psi}(f)}}$ for a positive definite field element f by sampling $\mathbf{q}_2 \leftarrow D_{\mathcal{O}_{n'},\mathbf{c}_2,\sqrt{\mathbf{\Psi}(d)}}$, then by updating the center $\mathbf{c}_1 := \mathbf{c}_1 + \mathbf{\Psi}(bd^{-1})(\mathbf{q}_2 - \mathbf{c}_2)$ and sampling $\mathbf{q}_2 \leftarrow D_{\mathcal{O}_{n'},\mathbf{c}_2,\sqrt{\mathbf{\Psi}(a-bd^{-1}b^*)}}$.

Similar to how SamplePz must sample $D_{\mathbb{Z}^{n\log q},\sqrt{s^2-\alpha^2}}$, the first step above requires sampling the discrete gaussian $D_{\mathcal{O}_{n'}^{\log q},\sqrt{s^2-\alpha^2}}$. This can be done in $O(\mathrm{rad}(n')n'\log q)$ since spherical discrete gaussians over the ring $\mathcal{O}_{n'}$ can be sampled in time $O(\mathrm{rad}(n')n')$ [41].

By using the randomized nearest plane algorithm to sample discrete gaussians on $\mathcal{O}_{n'}$ with diagonal covariances, sampling $\mathbf{p}$ statistically close to $D_{\mathcal{O}_{n'}^{2+\log q},\sqrt{\Sigma_p}}$ is $O(\mathrm{rad}(n')n'\log q) + \tilde{O}(n\log q)$. Note, the randomized rounder of [23, 47] could be used to sample $D_{\mathcal{O}_{n'},\sqrt{\mathbf{\Psi}(f)}}$. We conclude on observing that the above holds for any number field which has complex conjugation as an automorphism, though might not be as efficient because the ring/lattice of interest may have no sparse basis in the canonical embedding.

Acknowledgment

We thank Yuriy Polyakov, Kurt Rohloff, and Michael Walter for their helpful discussions as well as the anonymous reviewers for their helpful feedback and suggestions.

References

[1] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In H. Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.

[2] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Rabin [52], pages 98–115.

[3] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee. Functional encryption for threshold functions (or fuzzy IBE) from lattices. In Fischlin et al. [25], pages 280–297.

[4] M. Ajtai. Generating hard instances of the short basis problem. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.

[5] J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In Fischlin et al. [25], pages 334–352.

[6] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *Theory Comput. Syst.*, 48(3):535–553, 2011.

[7] R. E. Bansarkhani and J. A. Buchmann. Improvement and efficient implementation of a lattice-based signature scheme. In T. Lange, K. E. Lauter, and P. Lisonek, editors, *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, volume 8282 of *Lecture Notes in Computer Science*, pages 48–67. Springer, 2013.

[8] M. Bellare, E. Kiltz, C. Peikert, and B. Waters. Identity-based (lossy) trapdoor functions and applications. In Pointcheval and Johansson [50], pages 228–245.

[9] R. Bendlin, S. Krehbiel, and C. Peikert. How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE. In M. J. J. Jr., M. E. Locasto, P. Mohassel, and R. Safavi-Naini, editors, *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, volume 7954 of *Lecture Notes in Computer Science*, pages 218–236. Springer, 2013.

[10] D. Boneh and D. M. Freeman. Homomorphic signatures for polynomial functions. In K. G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 149–168. Springer, 2011.

[11] D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, volume 6571 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2011.

[12] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In P. Q. Nguyen and E. Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 533–556. Springer, 2014.

[13] X. Boyen. Attribute-based functional encryption on lattices. In *TCC*, pages 122–142, 2013.

[14] X. Boyen and Q. Li. Attribute-based encryption for finite automata from LWE. In M. H. Au and A. Miyaji, editors, *Provable Security - 9th International Conference, ProvSec 2015, Kanazawa, Japan, November 24-26, 2015, Proceedings*, volume 9451 of *Lecture Notes in Computer Science*, pages 247–267. Springer, 2015.

[15] Z. Brakerski and V. Vaikuntanathan. Circuit-abe from LWE: unbounded attributes and semi-adaptive security. In M. Robshaw and J. Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 363–384. Springer, 2016.

[16] Z. Brakerski, V. Vaikuntanathan, H. Wee, and D. Wichs. Obfuscating conjunctions under entropic ring LWE. In M. Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 147–156. ACM, 2016.

[17] J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In I. Visconti and R. D. Prisco, editors, *Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings*, volume 7485 of *Lecture Notes in Computer Science*, pages 57–75. Springer, 2012.

[18] R. Canetti and J. A. Garay, editors. *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*. Springer, 2013.

[19] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[20] G. Cetin, W. Dai, Y. Doroz, Y. Polyakov, K. Rohloff, H. Sajjadpour, E. Savas, and B. Sunar. Implementation and evaluation of lattice-based attributed-based encryption schemes. In preparation, (Personal communication), 2017.

[21] M. Clear and C. McGoldrick. Multi-identity and multi-key leveled FHE from learning with errors. In Gennaro and Robshaw [26], pages 630–656.

[22] D. B. Cousins, Y. Polyakov, K. Rohloff, G. Ryan, G. Sahu, H. Sajjadpour, and E. Savas. The palisade lattice crypto library. In preparation, (Personal communication), 2017.

[23] L. Ducas and P. Q. Nguyen. Faster gaussian lattice sampling using lazy floating-point arithmetic. In X. Wang and K. Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2012.

[24] L. Ducas and T. Prest. Fast fourier orthogonalization. In S. A. Abramov, E. V. Zima, and X. Gao, editors, *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016*, pages 191–198. ACM, 2016.

[25] M. Fischlin, J. A. Buchmann, and M. Manulis, editors. *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*. Springer, 2012.

[26] R. Gennaro and M. Robshaw, editors. *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*. Springer, 2015.

[27] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206. ACM, 2008.

[28] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Canetti and Garay [18], pages 75–92.

[29] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. *J. ACM*, 62(6):45:1–45:33, 2015. Prelim. version in STOC 2013.

[30] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Predicate encryption for circuits from LWE. In Gennaro and Robshaw [26], pages 503–523.

[31] S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In R. A. Servedio and R. Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 469–477. ACM, 2015.

[32] S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In M. Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 395–412. Springer, 2010.

[33] K. D. Gür, Y. Polyakov, K. Rohloff, G. W. Ryan, and E. Savaş. Implementation and evaluation of improved gaussian sampling for lattice trapdoors. Cryptology ePrint Archive, Report 2017/285, 2017. http://eprint.iacr.org/2017/285.

[34] K. D. Gur, Y. Polyakov, K. Rohloff, and E. Savas. Implementation of an improved lattice trapdoor and its evaluation for a digital signature scheme. In preparation, (Personal communication), 2017.

[35] P. N. Klein. Finding the closest lattice vector when it's unusually close. In D. B. Shmoys, editor, *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA.*, pages 937–941. ACM/SIAM, 2000.

[36] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In K. Sako and P. Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 41–61. Springer, 2013.

[37] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In H. Krawczyk, editor, *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, volume 8383 of *Lecture Notes in Computer Science*, pages 345–361. Springer, 2014.

[38] S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In K. Kurosawa and G. Hanaoka, editors, *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, volume 7778 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2013.

[39] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In K. Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, volume 5086 of *Lecture Notes in Computer Science*, pages 54–72. Springer, 2008.

[40] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43, 2013.

[41] V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-lwe cryptography. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2013.

[42] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[43] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Pointcheval and Johansson [50], pages 700–718.

[44] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[45] D. Micciancio and M. Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. Cryptology ePrint Archive, Report 2017/259, 2017. http://eprint.iacr.org/2017/259.

[46] P. Q. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In J. Katz, editor, *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, volume 9020 of *Lecture Notes in Computer Science*, pages 401–426. Springer, 2015.

[47] C. Peikert. An efficient and parallel gaussian sampler for lattices. In Rabin [52], pages 80–97.

[48] C. Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.

[49] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In D. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.

[50] D. Pointcheval and T. Johansson, editors. *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*. Springer, 2012.

[51] T. Pöppelmann, L. Ducas, and T. Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. In *Proceedings of the 16th International Workshop on Cryptographic Hardware and Embedded Systems — CHES 2014 - Volume 8731*, pages 353–370, New York, NY, USA, 2014. Springer-Verlag New York, Inc.

[52] T. Rabin, editor. *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*. Springer, 2010.

[53] H. Wee. Public key encryption against related key attacks. In Fischlin et al. [25], pages 262–279.

[54] F. Zhang. *The Schur Complement and Its Applications*, volume 4. Springer Science, 2006.