Discrete gaussian sampling for BKZ-reduced basis

Amaury Pouly[1][0000−0002−2549−951X] and Yixin Shen[2][0000−0002−8657−9337]

[1] Centre National de la Recherche Scientifique (CNRS) amaury.pouly@cnrs.fr
[2] Univ Rennes, Inria, CNRS, IRISA, Rennes yixin.shen@inria.fr

Abstract. Discrete Gaussian sampling on lattices is a fundamental problem in lattice-based cryptography. In this paper, we revisit the Markov chain Monte Carlo (MCMC)-based Metropolis-Hastings-Klein (MHK) algorithm proposed by Wang and Ling and study its complexity under the Geometric Series Assuption (GSA) when the given basis is BKZ-reduced. We give experimental evidence that the GSA is accurate in this context, and we give a very simple approximate formula for the complexity of the sampler that is accurate over a large range of parameters and easily computable. We apply our results to the dual attack on LWE of [21] and significantly improve the complexity estimates of the attack. Finally, we provide some results of independent interest on the Gaussian mass of a random q-ary lattices.

Keywords: Lattices · Discrete Gaussian Sampling · Geometric Series Assumption

1 Introduction

Discrete Gaussian sampling on lattices (DGS) is a fundamental problem in lattice-based cryptography. It appears both in basic cryptographic primitives such as "hash-and-sign" digital signature schemes [12,11], and in cryptanalysis as a fundamental tool for solving hard problems such as the Shortest Vector problem [3] or the Learning with Errors problem [21].

A Discrete Gaussian sampler is parameterized by a parameter "s" that controls the width of the distribution. In general, the smaller s is, the harder it is to construct the sampler. One important notion is called the smoothing parameter [19]. It captures the idea that sampling for a value of s above this threshold is significantly easier than sampling below because the distribution looks more like a continuous Gaussian in the former case.

There is currently a gap in the literature concerning discrete Gaussian samplers. We either have efficient but limited (s depends on the basis and must be large enough[3]) samplers [15,12,5,2] or very inefficient but arbitrarily good samplers [3]. The latter takes times $2^{n+o(n)}$. For certain applications such as dual attacks on LWE, it would be preferable to have access to a less rigid sampler that lies somewhere in-between, *i.e.* that can sample at any value of s and such

[3] Specifically, above a certain quantity that is always strictly greater than the smoothing parameter.

that the complexity smoothly interpolates between polynomial and exponential. Currently, the only known sampler to do that is the Monte Carlo Markov Chain-based algorithm of [28]. It works for all values of s but the complexity formula is involved and depends significantly the basis of the lattice. The authors gave a generic upper bound that does not depend on the shape of the basis but only applies to rather large values of s.

A natural question is whether we can obtain a better complexity bound for [28] when the basis follows a certain shape. This is the case for example when the basis is BKZ-reduced, a common occurrence in cryptanalysis.

In [21], the authors gave a simple approximation formula for the complexity of [28] when the basis is BKZ-reduced, assuming the Geometric Series Assumption (GSA) holds for the basis. Their formula also only applied to a limited range of values of s due to the imprecision of the approximation. Furthermore, [21] did not provide any experiments to compare the complexity of the algorithm when using a BKZ-reduced basis with the complexity when using the GSA.

In this paper, we give a more precise, yet still simple, formula for the complexity of [28] for a BKZ-reduced basis. Our formula is valid over a wider range of values of s than [21] and we do a detailed analysis of the precision of the formula. More precisely, we numerically show that our formula almost perfectly captures the complexity of [28] assuming the GSA. Furthermore, we conduct numerical experiments to compare the formula of [28] with a BKZ-reduced basis against the same formula using the GSA. We observe that the GSA provides a reasonably accurate complexity in this case. Finally, we update the complexity estimates of the dual attack proposed in [21] using our new formula, as well as other improvements in the code.

We also prove some results of independent interest on random lattices. Specifically, we give probability bounds that the Gaussian mass of a random q-ary lattice is close to 1. This quantity appears naturally when studying the smoothing parameter of lattices.

Organization of the paper Section 2 contains preliminary technical results. Section 3 provides an upper bound on the complexity of [28]. Section 4 studies this upper bound in the case where the basis is BKZ-reduced. Section 5 contains an application of our formula from Section 4 to refine the complexity estimates of the dual attacks of [21]. Finally, Section 6 gives some probabilistic bounds on the Gaussian mass of a random lattice.

2 Preliminaries

We denote vectors and matrices in bold case. We denote by $\mathbf{x}^T$ the transpose of the (column) vector $\mathbf{x}$, which is therefore a row vector. For any vector $\mathbf{x} \in \mathbb{R}^n$, we denote by $\|\mathbf{x}\|$ its Euclidean norm. For any finite set X, we denote by $\mathcal{U}(X)$ the uniform distribution over X. As usual, if P and Q are two probability distributions over X and Y respectively, we denote by PQ the product distribution over $X \times Y$. For any two distributions P and Q, we denote by $\mathrm{d_{TV}}(P, Q)$ the

statistical distance (or total variation distance) between P and Q. Recall that the exponential integral can be defined for any $x \geqslant 0$ by

$$E_1(x) = \int_1^\infty \frac{e^{-xt}}{t}\,\mathrm{d}\,t. \tag{1}$$

Furthermore, we also have for any $a, b > 0$ that

$$\int_a^b \frac{e^{-xt}}{t}\,\mathrm{d}\,t = E_1(at) - E_1(tb). \tag{2}$$

2.1 Lattices

We denote by $\widehat{L}$ the dual of a lattice $L \subset \mathbb{R}^n$ defined by

$$\widehat{L} = \{\mathbf{x} \in \mathrm{span}(L) : \forall \mathbf{y} \in L,\, \langle \mathbf{y}, \mathbf{x}\rangle \in \mathbb{Z}\}.$$

Let $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$ and q be a prime power. We say that a lattice L is a n-dimensional q-*ary lattice* if $q\mathbb{Z}^n \subseteq L \subseteq \mathbb{Z}^n$. Given a matrix $\mathbf{A} \in \mathbb{Z}^{n\times k}$, we consider the following n-dimensional q-ary lattices:

$$L_q(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{s} \in \mathbb{Z}^k,\, \mathbf{A}\mathbf{s} = \mathbf{x} \bmod q\right\},$$
$$L_q^\perp(\mathbf{A}) = \left\{\mathbf{x} \in \mathbb{Z}^n : \mathbf{A}^T\mathbf{x} = \mathbf{0} \bmod q\right\}.$$

It is well-know that for any q-ary lattice L, there exists $\mathbf{A}$ and $\mathbf{B}$ such that $L = L_q(\mathbf{A}) = L_q^\perp(\mathbf{B})$, and that $\widehat{L_q^\perp(\mathbf{A})} = \frac{1}{q}L_q(\mathbf{A})$. Furthermore $\det(L_q(\mathbf{A})) = q^{n-\mathrm{rk}\,\mathbf{A}} \geqslant q^{n-k}$ and therefore $\det(L_q^\perp(\mathbf{A})) = q^{\mathrm{rk}\,\mathbf{A}} \leqslant q^k$. Finally, since $\mathbb{Z}_q$ is a field, a random matrix $\mathbf{A}$ has full rank (equal to k) with probability at least $1 - kq^{k-1-n}$.

We refer the reader to [9], [29, Section 2.5.1], [20] or [21] for more details on those constructions and why these lattices play a crucial role in lattice-based cryptography, in particular because of the LWE problem.

2.2 Discrete Gaussian distribution

Let $n \in \mathbb{N}$ and $s > 0$. For any $\mathbf{x} \in \mathbb{R}^n$, we let $\rho_s(\mathbf{x}) \coloneqq e^{-\pi\|\mathbf{x}\|^2/s^2}$. We extend ρ_s to sets by $\rho_s(X) = \sum_{\mathbf{x}\in X}\rho_s(\mathbf{x})$ for any set X. We denote the *discrete Gaussian distribution* over a lattice $L \subset \mathbb{R}^n$ by $D_{L,s}(\mathbf{x}) = \frac{\rho_s(\mathbf{x})}{\rho_s(L)}$ for any $\mathbf{x} \in L$. We denote $D_{L,1}$ by D_L for simplicity. Given a vector $\mathbf{t} \in \mathbb{R}^n$,the shifted discrete Gaussian distribution over L is defined by $D_{L,s,\mathbf{t}}(\mathbf{x}) = \frac{\rho_s(\mathbf{x}-\mathbf{t})}{\rho_s(L-\mathbf{t})}$ for any $\mathbf{x} \in L$. It is well-known by the Poisson summation formula that for any lattice L and any $s > 0$,

$$\rho_{1/s}(\widehat{L}) = s^{-n}\rho_s(L).$$

We will also use the fact that for any $\mathbf{t} \in \mathbb{R}^n$, $\rho_s(\mathbf{t}+L) \leqslant \rho_s(L)$. See e.g. [27] for a good introduction on this topic.

In general, the smaller s is, the harder it is to construct a sampler for $D_{L,s}$. The notion of smoothing parameter [19] captures the idea that sampling for a valuer of s above this threshold is significantly easier than sampling below because the distribution looks more like a continuous Gaussian. Formally, for any $\varepsilon > 0$, the smoothing parameter of a lattice L is defined by

$$\eta_\varepsilon(L) = \inf\Big\{ s > 0 : \rho_{1/s}(\widehat{L}) \leqslant 1 + \varepsilon \Big\}.$$

There are many algorithms to sample above the smoothing parameter [15,12,5], including a time-space trade-off [2]. Sampling below the smoothing parameter is much more challenging and usually inefficient [3]. At the extreme, sampling for sufficiently small values of s allows one to solve the Shortest Vector problem (SVP) [3] which is known to be NP-hard under randomized reduction [4]. The Monte Carlo Markov Chain based algorithm of [28] works for all values of s but the complexity significantly depends on s and the shape of the basis. We give a short description of this algorithm in Section 2.3.

We will also make use of the following simple lemma:

Lemma 1. *Define, for any $s > 0$,*

$$\tilde{\rho}(s) = \begin{cases} 1 + 2e^{-\pi/s^2} & \text{if } s \leqslant 1, \\ s(1 + 2e^{-\pi s^2}) & \text{otherwise.} \end{cases}$$

Then $\tilde{\rho}$ is a continuously increasing function and for any $s > 0$ and $\varepsilon = 7 \times 10^{-6}$,

$$0 < \rho_s(\mathbb{Z}) - \tilde{\rho}(s) < 2 \sum_{k=2}^{\infty} e^{-\pi k^2} \leqslant \varepsilon.$$

Proof. The continuity is immediate since $\lim_{s \to 1, s>1} \tilde{\rho}(s) = 1 + 2e^{-\pi} = \tilde{\rho}(1)$. It is clearly increasing over $(0, 1]$ so by continuity it suffices to show that it is increasing over $(1, \infty)$. To see that, note that the derivative over this interval is $1 + 2e^{-\pi s^2} - 4s^2\pi e^{-\pi s^2}$ which can easily be seen to be positive for all $s > 1$.

Over the interval $(0, 1]$, it is clear that $\rho_s(\mathbb{Z}) - \tilde{\rho}(s) = 2\sum_{k=2}^{\infty} e^{-\pi k^2/s^2}$ is increasing. Similarly over $(1, \infty)$, by the Poisson summation formula, it is clear that $\rho_s(\mathbb{Z}) - \tilde{\rho}(s) = 2s\sum_{k=2}^{\infty} e^{-\pi k^2 s^2}$ is decreasing. Therefore, by continuity, the maximum between $\rho_s(\mathbb{Z})$ and $\tilde{\rho}(s)$ is attained at $s = 1$.

2.3 The Metropolis-Hastings-Klein (MHK) algorithm

In [28], the authors analyze a Markov chain Monte Carlo (MCMC)-based sampling algorithm called the independent Metropolis-Hastings-Klein (MHK) algorithm. Without going into the details, the Metropolis-Hastings algorithm is a particular way of sampling from a distribution which can be defined as the stationary distribution of an associated Markov chain. This algorithm is very flexible and requires to choose a "proposal distribution" which affects the speed of convergence of the Markov chain. In the particular case of the lattice discrete

Gaussian distribution, the authors in [28] use the Klein algorithm [15] to define the proposal distribution and call this the MHK algorithm. In a previous paper, the authors had already shown that the associated Markov chain converges exponentially quickly to the stationary distribution. The main contribution of [28] is then to analyze the spectral gap of the transition matrix of the associated Markov chain. This spectral gap is what defines the rate of convergence of the chain and therefore the mixing time which defines the number of steps of the algorithm. Note that by design, this algorithm always samples with an error since the chain converges to, but does not attain, its stationary distribution: by increasing the number of steps, we can nevetherless get closer to it in total variation. Finally, the algorithm only performs elementary matrix and vector operations which take time polynomial in the dimension.

Theorem 1 ([28, Theorem 1, (8), (23) and (24)[4]). *There is an algorithm that given a basis of a lattice $L \subset \mathbb{R}^d$, any vector $\mathbf{t} \in \mathbb{R}^n$, any $\varepsilon > 0$ and any $s > 0$, returns a sample according to some distribution $\mathcal{D}_{L,s,\mathbf{t},\varepsilon}$ such that $\mathrm{d_{TV}}(\mathcal{D}_{L,s,\mathbf{t},\varepsilon}, D_{L,s,\mathbf{t}}) \leqslant \varepsilon$. This algorithm runs in time $\ln\left(\frac{1}{\varepsilon}\right) \cdot \frac{1}{\Delta} \cdot \mathsf{poly}(d)$ where $\frac{1}{\Delta} = \frac{1}{\rho_s(\mathbf{t}+L)} \prod_{i=1}^{k} \rho_{s/\|\widetilde{\mathbf{b}}_i\|}(\mathbb{Z})$ and $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_k$ are the Gram-Schmidt vectors of the basis.*

2.4 Random q-ary lattices

We will consider the distributions $\mathcal{L}_{n,k,q}$ and $\mathcal{L}^{\perp}_{n,k,q}$ of q-ary lattices defined over the set of integer lattices by

$$\mathcal{L}_{n,k,q}(L) = \Pr_{\mathbf{A} \sim \mathcal{U}(\mathbb{Z}_q^{n \times k})}[L = L_q(\mathbf{A})],$$
$$\mathcal{L}^{\perp}_{n,k,q}(L) = \Pr_{\mathbf{A} \sim \mathcal{U}(\mathbb{Z}_q^{n \times (n-k)})}\left[L = L_q^{\perp}(\mathbf{A})\right].$$

In other words, the distribution is obtained by taking a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$ with uniform and independently distributed entries, and looking at the q-ary lattice generated by $\mathbf{A}$; and similarly for the orthogonal version. When neither k nor $n-k$ are too small, those two distributions are very close [21, Lemma 5].

Those distributions satisfy good uniformity properties when q goes to infinity. In particular, the following theorem shows that we can compute statistical properties of lattices sampled according to $\mathcal{L}^{\perp}_{n,k,q}$. See [21, Section 2.5] for more context.

Theorem 2 ([21, Theorem 3]). *Let $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$ and q be a prime power. Let $1 \leqslant p$ and $f : (\mathbb{Z}^n)^p \to \mathbb{R}$, then*

$$\mathbb{E}_{L \sim \mathcal{L}^{\perp}_{n,k,q}}\left[\sum_{\mathbf{x}_1,\ldots,\mathbf{x}_p \in L} f(\mathbf{x}_1, \ldots, \mathbf{x}_p)\right] = \sum_{\mathbf{x}_1,\ldots,\mathbf{x}_p \in \mathbb{Z}^n} q^{(k-n)r(\mathbf{x}_1,\ldots,\mathbf{x}_p)} f(\mathbf{x}_1, \ldots, \mathbf{x}_p)$$

where $r(\mathbf{x_1}, \ldots, \mathbf{x_p}) := \mathrm{rk}_{\mathbb{Z}_q^n}(\mathbf{x}_1, \ldots, \mathbf{x}_p)$ is the rank of the $\mathbf{x}_i \bmod q$ over $\mathbb{Z}_q^n$.

[4] [28] uses the normal distribution $e^{-\|\mathbf{x}\|^2/2\sigma^2}$ so $s = \sqrt{2\pi}\sigma$ with our notations.

In this paper, we will only make use of the following special case to compute the variance of a sum over a lattice.

Corollary 1. *Let $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$ and q be a prime power. For any $f : \mathbb{Z}^n \to \mathbb{R}$,*

$$\mathbb{V}_{L \sim \mathcal{L}^{\perp}_{n,k,q}}\left[\sum_{\mathbf{x} \in L} f(\mathbf{x})\right] = (q^{k-n} - q^{2(k-n)}) \sum_{\mathbf{x} \in \mathbb{Z}^n \setminus q\mathbb{Z}^n} \sum_{\mathbf{u} \in q\mathbb{Z}^n} \sum_{\alpha \in \mathbb{Z}_q \setminus \{0\}} f(\mathbf{x}) f(\alpha \mathbf{x} + q\mathbf{u}).$$

Proof. Observe that

$$\begin{aligned}
\mathbb{V}_L\left[\sum_{\mathbf{x} \in L} f(\mathbf{x})\right] &= \mathbb{E}_L\left[\sum_{\mathbf{x},\mathbf{y} \in L} f(\mathbf{x}) f(\mathbf{y})\right] - \mathbb{E}_L\left[\sum_{\mathbf{x} \in L} f(\mathbf{x})\right]^2 \\
&= \sum_{\mathbf{x},\mathbf{y} \in \mathbb{Z}^n} q^{(k-n)\,\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x},\mathbf{y})} f(\mathbf{x}) f(\mathbf{y}) - \left(\sum_{\mathbf{x} \in \mathbb{Z}^n} q^{(k-n)\,\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x})} f(\mathbf{x})\right)^2 \\
&= \sum_{\mathbf{x},\mathbf{y} \in \mathbb{Z}^n} \left(q^{(k-n)\,\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x},\mathbf{y})} - q^{(k-n)(\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x}) + \mathrm{rk}_{\mathbb{Z}^n}\,\mathbf{y}))}\right) f(\mathbf{x}) f(\mathbf{y}).
\end{aligned}$$

We now look at the various cases:

- If $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x},\mathbf{y}) = 0$ then $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x}) = \mathrm{rk}_{\mathbb{Z}^n}(\mathbf{y}) = 0$ so those terms of the sum are 0.
- If $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x},\mathbf{y}) = 2$ then $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x}) = \mathrm{rk}_{\mathbb{Z}^n}(\mathbf{y}) = 1$ so those terms of the sum are 0.
- If $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x},\mathbf{y}) = 1$ and $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x}) = 0$ then $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{y}) = 1$ so those terms of the sum are 0.
- The same holds if $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x},\mathbf{y}) = 1$ and $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x}) = 1$.

Therefore the only potentially non-zero terms are those for which $\mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x},\mathbf{y}) = \mathrm{rk}_{\mathbb{Z}^n}(\mathbf{x}) = \mathrm{rk}_{\mathbb{Z}^n}(\mathbf{y}) = 1$. When this is the case, this means that $\mathbf{x} \in \mathbb{Z}^n \setminus q\mathbb{Z}^n$ and there exists $\alpha \in \mathbb{Z}_q \setminus \{0\}$ such that $\mathbf{y} = \alpha\mathbf{x} \bmod 0$, *i.e.* $\mathbf{y} = \alpha\mathbf{x} + q\mathbf{u}$ for some $\mathbf{u} \in q\mathbb{Z}^n$.

2.5 BKZ

The BKZ algorithm is a well-known lattice reduction algorithm [25]. It processes the basis in blocks of size β and achieves a trade-off between the reduction quality and the running time. We refer the reader to [13] or [16] to recent work on this topic.

Let $\mathbf{B}$ be a BKZ-β reduced basis of a rank k lattice in $\mathbb{R}^d$ and $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_k$ be the corresponding Gram-Schmidt vectors. First recall that the root Hermite factor $\delta_{\mathbf{B}}$ is defined by

$$\|\mathbf{b}_1\| = \delta_{\mathbf{B}}^{k-1} \,\mathrm{vol}(L)^{1/k}.$$

By [13], we have that $\delta_{\mathbf{B}}^m \leqslant 2\gamma_\beta^{\frac{m-1}{2(\beta-1)}+\frac{3}{2}}$ where γ_β is the β-Hermite number. Experimentally, it has been verified [6] that

$$\delta_{\mathbf{B}} \approx H_\beta \coloneqq \left(\frac{\beta}{2\pi e}(\pi\beta)^{1/\beta}\right)^{1/2(\beta-1)} \tag{3}$$

See [10] for more details on this point. We also need to estimate $\|\widetilde{\mathbf{b}}_i\|$. For this, we will assume that the Geometric Series Assumption (GSA) [24] holds for BKZ-β reduced basis.

Heuristic 1 (Geometric Series Assumption (GSA)). *Let $\mathbf{b}_1, \ldots, \mathbf{b}_m$ be a BKZ-β reduced basis and $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_m$ be the corresponding Gram-Schmidt vectors. Then for all $i = 1, \ldots, m$,*

$$\|\widetilde{\mathbf{b}}_i\| = \|\mathbf{b}_1\| H_\beta^{-2(i-1)}, \qquad \|\mathbf{b}_1\| = H_\beta^{k-1}\,\mathrm{vol}(L)^{1/k}.$$

The GSA is known to be reasonably accurate when $\beta \ll m$ and $\beta \gg 50$ which is the case in our experiments, but it does not correctly model what happens in the last $m - \beta$ coordinates. See [1] for detailed discussions on the shape of the BKZ-reduced basis, and a more thorough literature review on this topic.

3 Complexity of DGS

The complexity of the sampling algorithm (Theorem 1) from [28] primarily depends on the quantity

$$\frac{1}{\Delta} = \frac{1}{\rho_s(\mathbf{t}+L)} \prod_{i=1}^{k} \rho_{s/\|\widetilde{\mathbf{b}}_i\|}(\mathbb{Z}). \tag{4}$$

Estimating this quantity is not easy because it depends on all the $\widetilde{\mathbf{b}}_i$, and on $\rho_s(\mathbf{t}+L)$. As was previously observed in [28], we can find an upper bound on this quantity that is quite tight when s is not too small and $\mathbf{t} = 0$ (or s is above the smoothing the parameter).

Lemma 2. *For any $s > 0$, lattice L and $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_k$ the Gram-Schmidt vectors of a basis of L,*

$$\frac{1}{\rho_s(L)} \prod_{i=1}^{k} \rho_{s/\|\widetilde{\mathbf{b}}_i\|}(\mathbb{Z}) = \frac{1}{\rho_{1/s}(\widehat{L})} \prod_{i=1}^{k} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z}) \leqslant \prod_{i=1}^{k} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})$$

Remark 1. When $\mathbf{t} \neq 0$ in (4), we cannot apply Lemma 2 directly. This is because for certain choices of $\mathbf{t}$, s and L, we might have $\rho_s(\mathbf{t}+L) < 1$. In this case, as was already noted in [28, above (76)], we can at least give a bound when s is above the smoothing parameter of the lattice. Indeed, if $s \geqslant \eta_\varepsilon(L)$ then $\frac{1}{\rho_s(\mathbf{t}+L)} \leqslant \frac{1+\varepsilon}{1-\varepsilon}\frac{1}{\rho_s(L)}$ by [22, Claim 3.8]. In this paper, we will only be interested in the case $\mathbf{t} = 0$.

Proof. Recall the standard fact that $\mathrm{vol}(L) = \prod_{i=1}^{k} \|\widetilde{\mathbf{b}}_i\|$. Using the Poisson summation formula, we get that

$$\frac{\prod_{i=1}^{k} \rho_{s/\|\widetilde{\mathbf{b}}_i\|}(\mathbb{Z})}{\rho_s(L)} = \frac{\prod_{i=1}^{k} \frac{s}{\|\widetilde{\mathbf{b}}_i\|} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})}{\frac{s^k}{\mathrm{vol}(L)} \rho_{1/s}(\widehat{L})}$$
$$= \frac{\mathrm{vol}(L)}{\prod_{i=1}^{k} \|\widetilde{\mathbf{b}}_i\|} \frac{\prod_{i=1}^{k} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})}{\rho_{1/s}(\widehat{L})} = \frac{\prod_{i=1}^{k} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})}{\rho_{1/s}(\widehat{L})}$$

and we get the wanted inequality since $\rho_{1/s}(\widehat{L}) \geqslant 1$.

This upper bound (the last inequality of Lemma 2) is more convenient to study since it does not depend on $\rho_s(L)$. On the other hand, we need to keep in mind that it is only tight when $\rho_{1/s}(\widehat{L}) \approx 1$ or at least $\rho_{1/s}(\widehat{L})$ is not large. This is precisely the definition of the smoothing parameter. For example, we might only want to use Lemma 2 for $s \geqslant \eta_1(L)$ to guarantee that $\rho_{1/s}(\widehat{L}) \leqslant 2$. Unfortunately, estimating η_1 is difficult for arbitrary lattices [7] and the generic bounds are very pessimistic.

In practice, however, we will most likely apply the sampling algorithm to random lattices. In this case, we can hope to obtain bounds on $\rho_{1/s}(\widehat{L})$ for most lattices. This is exactly what we do in Section 6 for random q-ary lattices which are fundamental for LWE-based cryptography.

q-ary lattices By Corollary 2, for any $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$, prime power q, $\xi > 1$ and α, if $s = \xi q^{k/n}$, $q^{k/n} \geqslant 2$ and $\alpha > \mu$ then

$$\Pr_{L \sim \mathcal{L}^{\perp}_{n,k,q}} \left[\rho_{1/s}(\widehat{L}) > \alpha \right] \leqslant \frac{\sigma^2}{(\alpha - \mu)^2}$$

where

$$\mu = 1.000007^n + \xi^{-n} \cdot 1.000014^n, \qquad \sigma^2 = q \cdot 1.000028^n \cdot \xi^{-n}.$$

If we assume that $n \leqslant 10000$, which is always true in practice, then all the above constants are very close to 1 and for $\alpha = 2$, we get that

$$\Pr_{L} \left[\rho_{1/s}(\widehat{L}) > 2 \right] \leqslant A \cdot \xi^{-n}$$

for some small constant A. Furthermore, a random lattice $L \sim \mathcal{L}^{\perp}_{n,k,q}$ satisfies that $\mathrm{vol}(L) = q^k$ with overwhelming probability. When this is the case, $s = \xi \,\mathrm{vol}(L)^{1/n}$. If we take $\xi = 1.1$ for example, then $\rho_{1/s}(\widehat{L}) > 2$ with overwhelming probability for large values of n.

Summary We can estimate that as soon as $s \geqslant \mathrm{vol}(L)^{1/n}$ then we essentially have $\rho_{1/s}(\widehat{L}) \leqslant 2$ with overwhelming probability over the choice of L, for large enough values of n.

4 DGS for BKZ-reduced basis

The goal of this section is to study the complexity of the sampler given by Theorem 1 when the basis is BKZ-reduced. More precisely, we will study the upper bound in Lemma 2:

$$\prod_{i=1}^{k} \rho_{\|\tilde{\mathbf{b}}_i\|/s}(\mathbb{Z}). \tag{5}$$

Recall that for values of s that are not tool small, this upper bound is quite tight (see previous section).

4.1 How accurate is the GSA?

In this section, we compare the values given by (5) when using actual BKZ-reduced basis or when using the GSA (Heuristic 1) for the values of the $\|\tilde{\mathbf{b}}_i\|$. We will refer to the former by "(5)+BKZ" and to the latter by "(5)+GSA".

Before going into the experimental results, it is useful to heuristically think about why the GSA should give good results in this context. Recall that the GSA is known to be quite accurate for most lattices, except in the head and in the tail. Looking at (5), we can expect that for values of s that are not too small, all terms of the product will be very close to 1. Since the GSA is accurate for most values, the only errors will come from a few terms in the head and in the tail. But since those terms are close to 1, we expect the overall error of (5)+GSA to be small.

We run the following experiment: for several values of $k = n$ and β, we pick $N = 5$ bases at random and BKZ-β reduce them. We then plot the complexity given by (5)+BKZ for each of those N bases. On the other hand, we also plotted the value given by (5)+GSA. Since the latter only depends on $x = \mathrm{vol}(L)^{1/n}/s$ (for fixed n and β), we plot all curves as a function of x. As discussed in Section 4, the upper bound (5) is only tight for values of s that satisfy $s \gtrapprox \mathrm{vol}(L)^{1/k}$, *i.e.* $x \leqslant 1$. Therefore we only plot the curves over the interval $[0, 1]$. To make the comparison easier, we give two plots per value of k and β:

- the "upper" plot gives the (logarithm) of (5)+GSA in red and the (N values of) (5)+BKZ in grey,
- the "lower" plot gives the (N values of) of the (logarithm) of $\frac{(5)+\mathrm{BKZ}}{(5)+\mathrm{GSA}}$ in blue.

In certain applications, it is important to run the sampler on q-ary bases. It is well-known [1,8] that running BKZ on the standard[5] q-ary basis yields a basis of a very particular shape called the "Z-shape". The Z-shape can deviate substantially from the GSA for certain choices of parameters n, k and q and it is still an open problem to give a good model for those bases. For this reason, we also ran the same experiments with some q-ary bases. Strangely, in our experiments, we

[5] A basis of the form $\begin{bmatrix} I_r & 0 \\ \mathbf{B} & qI_{n-r} \end{bmatrix}$ for some $1 \leqslant r \leqslant n$ and integer matrix $\mathbf{B}$.

observed that the GSA seems to give better results than the Z-shape adapted GSA, which is why in Figure 2 we plot (5)+GSA. We leave as an open question to explain why this is the case.

The results can be found in Figure 1 and Figure 2. We observe a reasonably good agreement between (5)+BKZ and (5)+GSA. Unsurprisingly, the error increases as s becomes smaller (and x becomes closer to 1) but we expect that most applications of this result will only use small values of x. In particular, the error seems negligible when $x \leqslant 1/4$ which is probably the more useful regime for this algorithm. In particular, our application in Section 5 only requires values of x which are significantly smaller than $1/4$.

4.2 An approximation formula

Having observed in the previous section that the GSA gives reasonably accurate values for (5), we now give a simple approximation for it. The motivation is twofold. First, from a theoretical perspective, it is difficult to understand the behaviour of (5), even assuming the GSA. By finding a much simpler formula, we can better understand its dependency on the various parameters. Second, when using (5) in an optimizer to compute complexity estimates of attacks (such as in [21]), the cost of evaluating this formula can quickly become prohibitive. Indeed, evaluating (5) takes time $O(k)$ to evaluate, compared to $O(1)$ to the formula that we give.

Theorem 3. *Let $0 < k < n$ and $0 < \beta \leqslant k$. Let $\mathbf{b}_1, \ldots, \mathbf{b}_k \in \mathbb{R}^n$ be a BKZ-β reduced basis of a lattice L and $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_k$ be the Gram-Schmidt vectors of the basis. Let $s > 0$ and $\alpha = \|\mathbf{b}_1\|/s$. If Heuristic 1 holds and $H_\beta > 1$ then*

$$\ln\left(\prod_{i=1}^{k} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})\right) \approx A + B + C$$

where

$$\begin{aligned} A &= (k_0+1)\ln(\alpha) - k_0(k_0+1)\ln(H_\beta), \\ B &= \frac{E_1\left(\pi\alpha^2 H_\beta^{-4(k_0+1)}\right) - E_1(\pi\alpha^2)}{2\ln(H_\beta)}, \\ C &= \frac{E_1\left(\pi\alpha^{-2} H_\beta^{4(k_0+1)}\right) - E_1\left(\pi\alpha^{-2} H_\beta^{4k}\right)}{2\ln(H_\beta)} \end{aligned}$$

where $k_0 = \max\left(-1, \min\left(k-1, \left\lfloor \frac{\ln(\alpha)}{2\ln(H_\beta)} \right\rfloor\right)\right)$.

Remark 2. We have experimentally observed that B and C are usually much smaller than A. Nevertheless, B and C significantly increase the precision of the formula for larger values of s. Ignoring B and C for a moment, we see that A "smoothly" interpolates between two extremes:

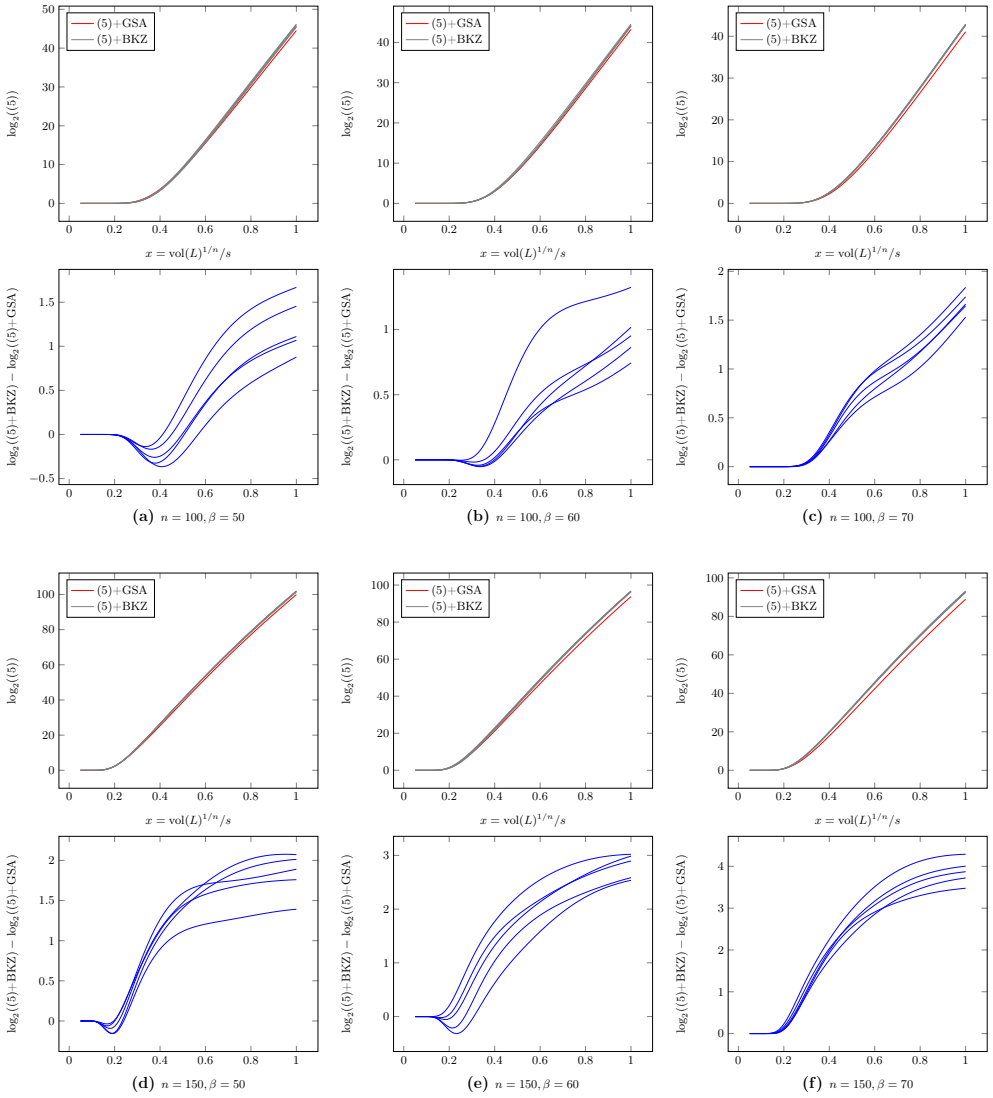
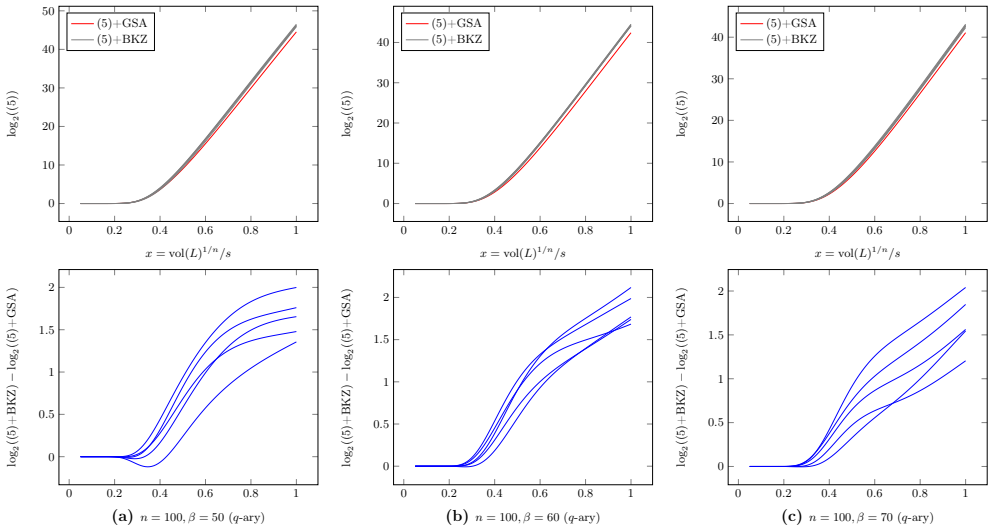


Fig. 1. Comparison between (5)+BKZ and (5)+GSA for various values of n and β. For each experiment, $N = 5$ bases are chosen at random and BKZ-β reduced. The plots show both the absolute values and the ratio between the two complexities. See Section 4.1 for details.



(a) $n = 100, \beta = 50$ (q-ary) (b) $n = 100, \beta = 60$ (q-ary) (c) $n = 100, \beta = 70$ (q-ary)

Fig. 2. Comparison between (5)+BKZ and (5)+GSA for various values of n and β. For each experiment, $N = 5$ q-ary basis are chosen at random and BKZ-β reduced. The plots show both the absolute values and the ratio between the two complexities. See Section 4.1 for details.

- When $k_0 = -1$ (small values of α, *i.e.* large values of s), $A = 0$ so the complexity bound is constant.
- When $k_0 = k-1$ (large values of α, *i.e.* small values of s), then we must have $\ln(\alpha) \geqslant (k-1)2\ln H_\beta$ and therefore the complexity is at least $\frac{k}{2}\ln\alpha$. Since $\alpha = \|\mathbf{b}_1\|/s$ and $\mathbf{b}_1 = H_\beta^{k-1}\,\mathrm{vol}(L)^{1/k}$ by the GSA, then the complexity is at least $\frac{k^2}{2}\ln H_\beta + \ln \mathrm{vol}(L)$. Since, very roughly, $\ln H_\beta = \frac{1}{\beta}\ln\beta$, the complexity of the algorithm will already be more than 2^n, by which point this sampler becomes worse than that of [3].

Proof (Proof of Theorem 3). Using Heuristic 1, we have that $\|\widetilde{\mathbf{b}}_i\|/s \approx \alpha H_\beta^{-2(i-1)}$. Therefore,

$$\ln\left(\prod_{i=1}^{k}\rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})\right) \approx \sum_{i=0}^{k-1}\ln\rho_{\alpha H_\beta^{-2i}}(\mathbb{Z})$$

Now check that

$$\alpha H_\beta^{-2i} \geqslant 1 \quad \Leftrightarrow \quad i \leqslant \frac{\ln(\alpha)}{2\ln(H_\beta)}.$$

We let

$$k_0 = \max\left(-1, \min\left(k-1, \left\lfloor\frac{\ln(\alpha)}{2\ln(H_\beta)}\right\rfloor\right)\right).$$

Then by Lemma 1 we have

$$\sum_{i=0}^{k-1} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) \approx \sum_{i=0}^{k_0} \ln\left(\alpha H_\beta^{-2i} \cdot \left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right)\right) + \sum_{i=k_0+1}^{k-1} \ln\left(1 + 2\exp(-\tfrac{\pi}{\alpha^2} H_\beta^{4i})\right)$$

One easily checks that

$$\sum_{i=0}^{k_0} \ln\left(\alpha H_\beta^{-2i}\right) = (k_0+1)\ln(\alpha) - k_0(k_0+1)\ln(H_\beta).$$

For any $a \leqslant b$ integers, $y > 0$ and $x \neq 1$, we have

$$\begin{aligned}
\sum_{i=a}^{b} \ln\left(1 + 2\exp(-yx^{4i})\right) &\approx 2\sum_{i=a}^{b} \exp(-yx^{4i}) && (6)\\
&\approx 2\int_a^{b+1} \exp(-yx^{4t})\,\mathrm{d}\,t && (7)\\
&= 2\int_{x^{4a}}^{x^{4(b+1)}} \frac{\exp(-yu)}{4\ln(x)u}\,\mathrm{d}\,u \quad \text{by the change } u = x^{4t}\\
&= \frac{E_1(yx^{4a}) - E_1(yx^{4(b+1)})}{2\ln(x)} \quad \text{by (2)}.
\end{aligned}$$

Therefore,

$$\sum_{i=0}^{k_0} \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right) \approx \frac{E_1\left(\pi\alpha^2 H_\beta^{-4(k_0+1)}\right) - E_1(\pi\alpha^2)}{2\ln(H_\beta)}$$

and

$$\sum_{i=k_0+1}^{k-1} \ln\left(1 + 2\exp(-\tfrac{\pi}{\alpha^2} H_\beta^{4i})\right) \approx \frac{E_1\left(\pi\alpha^{-2} H_\beta^{4(k_0+1)}\right) - E_1\left(\pi\alpha^{-2} H_\beta^{4k}\right)}{2\ln(H_\beta)}.$$

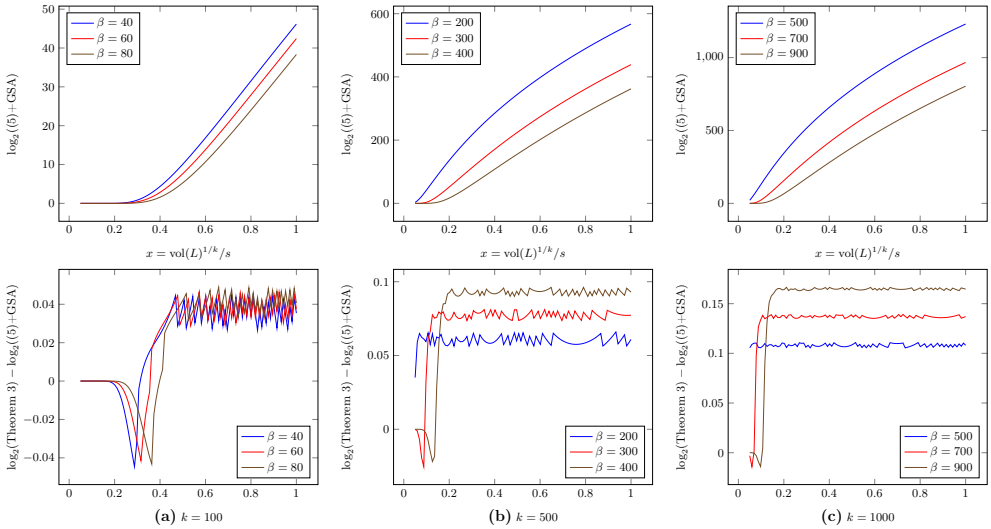
4.3 How accurate is the approximation?

We now compare the formula of Theorem 3 with the upper bound (5) on the complexity where we use the GSA (Heuristic 1) for the values of the $\|\tilde{\mathbf{b}}_i\|$. We will refer to the latter by "(5)+GSA" as we did in Section 4.1.

We observe that both (5)+GSA and the formula from Theorem 3 only depend on k, $x = \mathrm{vol}(L)^{1/k}/s$ and β. Therefore, we plot the complexity curves as a function of x. We will plot all results in logarithimic scale (base 2) since this is the most relevant scale for our applications. For each set of parameter, we plot both the absolute values and the difference. As discussed in Section 4, the upper bound (5) is only tight for values of s that satisfy $s \gtrapprox \mathrm{vol}(L)^{1/k}$, *i.e.* $x \leqslant 1$. Therefore we only plot the curves over the interval $[0, 1]$.

The curves can be found in Figure 3. The bottom figures confirm that the difference between Theorem 3 and (5)+GSA is minimal. Indeed, we can see that up to $k = 1000$, the logarithm of the ratio between the two quantities is less than 0.16, meaning that the approximation is correct within a multiplicative factor 1.1. This factor should be negligible for virtually all applications given that the complexities grows exponentially in k, as can be seen on the top figures.

We observe in Figure 3 that as k increases, the error between the approximation formula and the (5)+GSA seems to increase slowly and be almost constant over the range of x (except for very small values of x). We do not have a clear explanation for this phenomenon which could be due to the approximation made in the proof either at step (6) or (7).



(a) $k = 100$ **(b)** $k = 500$ **(c)** $k = 1000$

Fig. 3. Top pictures: (logarithm) of the complexity upper bound given by (5)+GSA, for different values of k and β, plotted as a function of x. Bottom pictures: (logarithm) of the ratio between the complexity given by Theorem 3 and that given by (5)+GSA, for the same values of k and β.

5 Applications to dual attack on LWE

In this section, we revisit the complexity estimates from [21] using our approximation formula (Theorem 3). The approach in [21] is to write an optimizer that uses an approximate formula to find the best parameters and to then re-evaluate the complexity for the best parameters using (5)+GSA. Indeed, recall that (5)+GSA takes time $O(n)$ to compute (compared to $O(1)$ for the approximation) which becomes prohibitive when $n \approx 1000$ in the dual attack. However,

this strategy can lead to sub-optimal parameter choices if the approximate formula for the sampler is not good enough.

5.1 High-level overview of the attack

In this section, we give a succint presentation of the attack in [21]. We focus on the high-level description and how the Gaussian sampler plays a role. In this attack, we are given m LWE samples which we represent in matrix form by $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is chosen uniformly at random, and $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ where $\mathbf{s} \in \mathbb{Z}_q^n$ is the unknown secret that we are trying to recover, and $\mathbf{e} \in \mathbb{Z}_q^m$ has its components sampled independently from a distribution χ_e. Typically χ_e will either be a modular discrete Gaussian, or a centered binomial. In all applications, χ_e will take very small values with high probability. Here, the number of samples m is a parameter of the attack and is typically around $2n$, see [21, Sections 4.4 and 7] for more discussion on this point.

The first step of the attack is to split the secret $\mathbf{s}$ into two parts $\mathbf{s}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and $\mathbf{s}_{\text{dual}} \in \mathbb{Z}_q^{n_{\text{dual}}}$ where $n = n_{\text{guess}} + n_{\text{dual}}$. The matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is correspondingly split into two parts:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\text{guess}} & \mathbf{A}_{\text{dual}} \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} \mathbf{s}_{\text{guess}} \\ \mathbf{s}_{\text{dual}} \end{bmatrix}.$$

The algorithm will now exhaustively try all values $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and check which one is correct. Check that

$$\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}} = \mathbf{A}_{\text{guess}} \cdot (\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}) + \mathbf{A}_{\text{dual}} \cdot \mathbf{s}_{\text{dual}} + \mathbf{e}.$$

Recall that the components of $\mathbf{e}$ are sampled from χ_e which is small, so we expect $\|\mathbf{e}\|$ to be relatively small. Consider the lattice

$$L_q(\mathbf{A}_{\text{dual}}) = \mathbf{A}_{\text{dual}} \mathbb{Z}_q^{n_{\text{dual}}} + \mathbb{Z}^m.$$

The intuition behind the attack is that:

- If $\mathbf{s}_{\text{guess}} = \tilde{\mathbf{s}}_{\text{guess}}$ then $\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}} \in L_q(\mathbf{A}_{\text{dual}}) + \mathbf{e}$ and since $\mathbf{e}$ has small norm, this means that $\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}$ is close to the lattice $L_q(\mathbf{A}_{\text{dual}})$.
- If $\mathbf{s}_{\text{guess}} \neq \tilde{\mathbf{s}}_{\text{guess}}$ then one can show that with high probability over the choice of $\mathbf{A}$, the vector $\mathbf{A}_{\text{guess}} \cdot (\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}})$ is far from the lattice $L_q(\mathbf{A}_{\text{dual}})$ and therefore $\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}$ is far from to the lattice $L_q(\mathbf{A}_{\text{dual}})$.

Therefore, the attack reduces to the problem of estimating the distance between a given vector $\mathbf{x}$ and the lattice $L_q(\mathbf{A}_{\text{dual}})$. The usual approach to do so is to first sample a large number N of vectors $\mathbf{w}_1, \ldots, \mathbf{w}_N$ in the dual lattice

$$L_q^{\perp}(\mathbf{A}_{\text{dual}}) = \left\{ \mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T \mathbf{A}_{\text{dual}} = \mathbf{0} \bmod q \right\}$$

according to a discrete Gaussian of width s (a parameter of the attack). We then consider the sum

$$g_W(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^{N} \cos(2\pi \langle \mathbf{x}, \mathbf{w}_j \rangle / q)$$

which can be shown to corrolate with the distance from $\mathbf{x}$ to $L_q(\mathbf{A}_{\text{dual}})$. Therefore it suffices to compute g_W for all guesses $\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}$ and to keep the highest one. While the naive way of computing all those sums is slow, a better algorithm using the discrete Fourier transform is possible.

A critical point in the analysis above is the number of samples N: it needs to be large enough for the values of g_W to correctly estimate the distance to $L_q(\mathbf{A})$ and how large depends on the width s of the discrete Gaussian according to which we sample the $\mathbf{w}_i$. Intuitively, a smaller value of s will require a smaller number of samples N, but will increase the complexity of the Gaussian sampler. Since [21] uses the sampler from [28], it is critical to have an accurate and quick to compute estimate of the complexity of the sampler given a width s.

5.2 Applications

Our approach is first to modify the code[6] to use our new approximate formula. This requires a few more changes since the optimizer of [21] enforces the condition[7] that $s \geqslant \frac{\|\mathbf{b}_1\|}{2q}$ [21, Section 4.4]. Furthermore, the optimizer of [21] always picks the smallest possible value of s. This approach does not work in our case because our condition $s \geqslant q^{k/n-1}$ is much weaker[8] than $\frac{\|\mathbf{b}_1\|}{2q}$, and results in very small values of s and sampling time which is too high. We instead modified the code to search for the value of s in the interval $\frac{\|\mathbf{b}_1\|}{q} \cdot [0.4, 0.5]$ which experimentally seems to give the best results. Our new complexity estimates are given in Table 1. We included the value of $x = q^{n_{\text{dual}}/m-1}/s$ in the table due make the correspondence with Section 4. Indeed, recall that the complexity of the sampler only depends on $x = \text{vol}(L)^{1/d}/t$ where d is the dimension of the lattice and t is the width of the discrete Gaussian. In the algorithm of [21], $d = m$, $\text{vol}(L) = q^{n_{\text{dual}}}$ and $t = qs$. Note that similarly to [21], we use the formula of Theorem 3 in the optimizer to find the best set of parameters but we compute the final estimates using (5)+GSA. Therefore, the only potential inaccuracies come from errors due to the GSA (see last paragraph of this section).

We observe some significant improvements in the complexity compared to [21], especially without modulus switching, thanks to the smaller values of s that our formula is able to handle. However, when looking in detail at the results, we also observe that the optimizer of [21] has some limitations. Indeed, the algorithm brute forces all possible values of m, β and n_{guess} but since the search space is too large, it only evaluates values on a grid with some signifcant steps on the β and n_{guess} axis. As a result, the various complexity terms (BKZ, guessing and sampling complexity) do not balance well in the final complexity and lead to sub-optimal results. This is why our second approach is to modify the optimizer to perform a coarse-grid search for promising parameter sets, and

[6] The code for the complexity estimates in [21] is available as an artifact.

[7] Beware that the algorithm of [21] actually samples at qs and not s.

[8] By the Gaussian heuristic, which essentially holds true for random q-ary lattices [21, Corollary 2], $\lambda_1 \approx q^{k/n}\sqrt{\frac{n}{2\pi e}}$. For a BKZ-$\beta$ reduced basis, $\|\mathbf{b}_1\| \geqslant \lambda_1$ and in fact $\|\mathbf{b}_1\| \gg \lambda_1$ unless β is close to n. Hence, $\frac{\|\mathbf{b}_1\|}{2} \gg q^{k/n}$ for most lattices.

then do a refined local search around those candidates. The results are available Table 2 and show much more signifcant improvements, including for estimates with modulus switching.

An interesting observation can be made on both Table 1 and Table 2: the values of x required for the sampler are all very small. Indeed, the largest value of x used by the algorithm is less than 0.01. Recall that in Section 4.1 we compared the complexity of the sampler BKZ-reduced basis against an approximation using the GSA. We saw a notable increase in the approximation error when x gets close to 1, but also a negligible error when $x \leqslant 0.2$. While it is difficult to extrapole results to dual attack (that use $\beta \approx 1000$) from limited experimental results ($\beta = 70$), we note that in all our experiments, the error was consistently negligible when $x \leqslant 0.2$. This suggests that in this parameter regime, we can hope that the complexity estimates are indeed accurate.

Table 1. Dual attack cost estimates and their parameters as described in [21, Section 4.4] modified as described in Section 5. All costs are logarithms in base two. Note that the cost of attacks with modulus switching are optimistic estimates of what an algorithm with modulus switching could give if the algorithm of [21] was extended with modulus switching. **This table only contains improvements on the sampler complexity.**

	No modulus switching							
Scheme	attack	m	n_{guess}	n_{dual}	β	s	x	attack [21]
Kyber512	182	963	15	497	541	0.200	0.097	185
Kyber768	267	1419	21	747	849	0.250	0.087	273
Kyber1024	366	1925	31	993	1202	0.250	0.079	376
	With modulus switching							
Kyber512	141	763	141	371	381	0.190	0.082	141
Kyber768	201	1119	201	567	599	0.240	0.077	202
Kyber1024	273	1575	261	763	867	0.240	0.064	279

6 On the Gaussian mass of random q-ary lattices

In this section, we give probabilistic estimates on the value of $\rho_{1/s}(\widehat{L})$ when L is a random q-ary lattice (see Section 2.4 for more details). These bounds are related to the smoothing parameter of lattices and are useful to argue about the tightness of the complexity bound in Section 3. We are not aware of any such results in the literature for these classes of random lattices. However, a closely related result is available in [14] which studies matrices with each entry independently and identically distributed from an integer Gaussian distribution.

Lemma 3. *For any $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$, prime power q and $s > 0$,*

$$\mathbb{E}_{L \sim \mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{1/s}(\widehat{L})\right] \leqslant \rho_{1/s}(\mathbb{Z}^n) + q^{k-n}\rho_{q/s}(\mathbb{Z}^n),$$

Table 2. Dual attack cost estimates and their parameters as described in [21, Section 4.4] modified as described in Section 5. All costs are logarithms in base two. Note that the cost of attacks with modulus switching are optimistic estimates of what an algorithm with modulus switching could give if the algorithm of [21] was extended with modulus switching. **This table contains improvements on the optimizer and the sampler.**

	No modulus switching							
Scheme	attack	m	n_{guess}	n_{dual}	β	s	x	attack [21]
Kyber512	181	1023	15	497	539	0.200	0.079	185
Kyber768	266	1504	22	746	843	0.240	0.070	273
Kyber1024	366	1985	31	993	1199	0.250	0.070	376
	With modulus switching							
Kyber512	136	778	133	379	381	0.190	0.081	141
Kyber768	199	1164	197	571	602	0.230	0.068	202
Kyber1024	270	1520	269	755	857	0.240	0.070	279

$$\mathbb{V}_{L\sim\mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{1/s}(\widehat{L})\right] \leqslant q^{1+k-n}\rho_{q/s}(\mathbb{Z}^n)\rho_{1/s}(\mathbb{Z}^n).$$

Proof. Recall that if $L = L_q(\mathbf{A})$ then $\widehat{L} = \frac{1}{q}L_q^{\perp}(\mathbf{A})$. Therefore, $L \sim \mathcal{L}_{n,k,q}$ is equivalent to $\widehat{L} \sim \frac{1}{q}\mathcal{L}^{\perp}_{n,k,q}$. Therefore we can use Theorem 2 to get that

$$\begin{aligned}
\mathbb{E}_{L\sim\mathcal{L}_{n,k,q}}\left[\rho_{1/s}(\widehat{L})\right] &= \mathbb{E}_{L\sim\mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{1/s}(\tfrac{1}{q}L)\right] \\
&= \mathbb{E}_{L\sim\mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{q/s}(L)\right] \\
&= \rho_{q/s}(q\mathbb{Z}^n) + q^{k-n}\rho_{q/s}(\mathbb{Z}^n \setminus q\mathbb{Z}^n) \\
&\leqslant \rho_{1/s}(\mathbb{Z}^n) + q^{k-n}\rho_{q/s}(\mathbb{Z}^n)
\end{aligned}$$

To estimate the variance, we use Corollary 1 to get that

$$\begin{aligned}
\mathbb{V}_{L\sim\mathcal{L}_{n,k,q}}\left[\rho_{1/s}(\widehat{L})\right] &= \mathbb{V}_{L\sim\mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{1/s}(\tfrac{1}{q}L)\right] \\
&= \mathbb{V}_{L}\left[\rho_{q/s}(L)\right] \\
&= (q^{k-n} - q^{2(k-n)}) \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus q\mathbb{Z}^n}\sum_{\mathbf{u}\in q\mathbb{Z}^n}\sum_{\alpha\in\mathbb{Z}_q\setminus\{0\}} \rho_{q/s}(\mathbf{x})\rho_{q/s}(\alpha\mathbf{x}+q\mathbf{u}) \\
&\leqslant q^{k-n} \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus q\mathbb{Z}^n}\sum_{\alpha\in\mathbb{Z}_q\setminus\{0\}} \rho_{q/s}(\mathbf{x})\rho_{q/s}(\alpha\mathbf{x}+q\mathbb{Z}^n) \\
&\leqslant q^{k-n} \sum_{\mathbf{x}\in\mathbb{Z}^n\setminus q\mathbb{Z}^n}\sum_{\alpha\in\mathbb{Z}_q\setminus\{0\}} \rho_{q/s}(\mathbf{x})\rho_{q/s}(q\mathbb{Z}^n) \\
&= (q-1)q^{k-n}\rho_{q/s}(\mathbb{Z}^n \setminus q\mathbb{Z}^n)\rho_{q/s}(q\mathbb{Z}^n) \\
&\leqslant q^{1+k-n}\rho_{q/s}(\mathbb{Z}^n)\rho_{1/s}(\mathbb{Z}^n).
\end{aligned}$$

Lemma 4. *For any $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$, prime power q and $\xi > 1$, if $s = \xi q^{k/n}$ then*

$$\rho_{1/s}(\mathbb{Z}^n) \leqslant (1+\varepsilon)^n f(s),$$
$$q^{k-n}\rho_{q/s}(\mathbb{Z}^n) \leqslant (1+\varepsilon)^n \xi^{-n} f\left(q^{1-k/n}/\xi\right)^n$$

where $f(x) = 1 + 2e^{-\pi x^2}$ for all $x \geqslant 0$ and ε is defined in Lemma 1.

Proof. We will show the result when $s \leqslant q$. When $s > q$ then the result follows trivially since ρ_s is an increasing function of s. Let ε be as in Lemma 1. Clearly $s \geqslant 1$ if $s = \xi q^{k/n}$ so we can apply Lemma 1 to get that

$$\rho_{1/s}(\mathbb{Z}^n) \leqslant (1+\varepsilon)^n \left(1 + 2e^{-\pi s^2}\right)^n = (1+\varepsilon)^n f(s).$$

By Lemma 1, when $s \leqslant q$, we have that

$$\begin{aligned} q^{k-n}\rho_{q/s}(\mathbb{Z}^n) &\leqslant q^{k-n}(1+\varepsilon)^n (\tfrac{q}{s})^n \left(1 + 2e^{-\pi (q/s)^2}\right)^n \\ &= q^{k-n}(1+\varepsilon)^n \xi^{-n} q^{n-k} \left(1 + 2e^{-\pi (q^{1-k/n}/\xi)^2}\right)^n \\ &= (1+\varepsilon)^n \xi^{-n} \left(1 + 2e^{-\pi (q^{1-k/n}/\xi)^2}\right)^n \\ &= (1+\varepsilon)^n \xi^{-n} f\left(q^{1-k/n}/\xi\right)^n. \end{aligned}$$

Corollary 2. *For any $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$, prime power q, $\xi > 1$ and α, if $s = \xi q^{k/n}$, $q^{k/n} \geqslant 2$ and $\alpha > \mu$ then*

$$\Pr_{L \sim \mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{1/s}(\widehat{L}) > \alpha\right] \leqslant \frac{\sigma^2}{(\alpha - \mu)^2}$$

where

$$\mu = 1.000007^n + \xi^{-n} \cdot 1.000014^n, \qquad \sigma^2 = q \cdot 1.000028^n \cdot \xi^{-n}.$$

Proof. We will show the result when $2s \leqslant q$. When $s > q$ then the result follows trivially since ρ_s is an increasing function of s.

Let f be defined as in Lemma 4 which is a decreasing function. Observe that if $2s \leqslant q$ then $2\xi qk/n \leqslant q$, that is $q^{1-k/n}/\xi \geqslant 2$. Therefore, $f(q^{1-k/n}/\xi) \leqslant f(2) \leqslant 1.000007$. Similarly, if $q^{k/n} \geqslant 2$ then $s \geqslant 2$ so $f(s) \leqslant f(2)$. Also note that for $\varepsilon = 7 \times 10^{-6}$ we have $(1+\varepsilon)f(2) \leqslant 1.000014$. Hence, by Lemma 3 and Lemma 4

$$\begin{aligned} \mu &:= \mathbb{E}_{L \sim \mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{1/s}(\widehat{L})\right] \\ &\leqslant \rho_{1/s}(\mathbb{Z}^n) + q^{k-n}\rho_{q/s}(\mathbb{Z}^n), \\ &\leqslant (1+\varepsilon)^n f(s)^n + (1+\varepsilon)^n \xi^{-n} f\left(q^{1-k/n}/\xi\right)^n \end{aligned}$$

$$\begin{aligned}
&\leqslant (1+\varepsilon)^n f(2)^n + (1+\varepsilon)^n \xi^{-n} f\left(2\right)^n \\
&\leqslant 1.000007^n + \xi^{-n} \cdot 1.000014^n,
\end{aligned}$$

and

$$\begin{aligned}
\sigma^2 &:= \mathbb{V}_{L \sim \mathcal{L}^{\perp}_{n,k,q}} \left[\rho_{1/s}(\widehat{L}) \right] \\
&\leqslant q^{1+k-n} \rho_{q/s}(\mathbb{Z}^n) \rho_{1/s}(\mathbb{Z}^n) \\
&\leqslant q \cdot (1+\varepsilon)^n \xi^{-n} f\left(q^{1-k/n}/\xi\right)^n \cdot (1+\varepsilon)^n f(s)^n \\
&\leqslant q \cdot (1+\varepsilon)^n \xi^{-n} f\left(2\right)^n \cdot (1+\varepsilon)^n f(2)^n \\
&\leqslant q \cdot 1.000028^n \cdot \xi^{-n}.
\end{aligned}$$

Finally, we conclude by Chebyshev's inequality.

The constants in Corollary 2 are somewhat arbitrary but allow for a greatly simplified statement. It seems that the probability bound is not very sharp and it would be interesting to see if the proof can be refined to obtain a stronger statement.

References

1. Lattice Attacks on NTRU and LWE: A History of Refinements, p. 15–40. London Mathematical Society Lecture Note Series, Cambridge University Press (2021)
2. Aggarwal, D., Chen, Y., Kumar, R., Shen, Y.: Improved (provable) algorithms for the shortest vector problem via bounded distance decoding. In: Bläser, M., Monmege, B. (eds.) 38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference). LIPIcs, vol. 187, pp. 4:1–4:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.STACS.2021.4
3. Aggarwal, D., Dadush, D., Regev, O., Stephens-Davidowitz, N.: Solving the shortest vector problem in 2^n time using discrete gaussian sampling: Extended abstract. In: Servedio, R.A., Rubinfeld, R. (eds.) Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015. pp. 733–742. ACM (2015). https://doi.org/10.1145/2746539.2746606
4. Ajtai, M.: The shortest vector problem in l2 is np-hard for randomized reductions (extended abstract). In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. p. 10–19. STOC '98, Association for Computing Machinery, New York, NY, USA (1998). https://doi.org/10.1145/276698.276705
5. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. p. 575–584. STOC '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2488608.2488680
6. Chen, Y.: Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. Ph.D. thesis (2013), http://www.theses.fr/2013PA077242, thèse de doctorat dirigée par Nguyen, Phong Q. Informatique Paris 7 2013

7. Chung, K.M., Dadush, D., Liu, F.H., Peikert, C.: On the lattice smoothing parameter problem. In: 2013 IEEE Conference on Computational Complexity. pp. 230–241 (2013). https://doi.org/10.1109/CCC.2013.31
8. Ducas, L., Espitau, T., Postlethwaite, E.W.: Finding short integer solutions when the modulus is small. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 150–176. Springer Nature Switzerland, Cham (2023)
9. Erez, U., Litsyn, S., Zamir, R.: Lattices which are good for (almost) everything. IEEE Transactions on Information Theory **51**(10), 3401–3416 (2005). https://doi.org/10.1109/TIT.2005.855591
10. Espitau, T., Joux, A., Kharchenko, N.: On a Dual/Hybrid Approach to Small Secret LWE: A Dual/Enumeration Technique for Learning with Errors and Application to Security Estimates of FHE Schemes. In: Bhargavan, K., Oswald, E., Prabhakaran, M. (eds.) Progress in Cryptology – INDOCRYPT 2020, vol. 12578, p. 440–462. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-65277-7_20, series Title: Lecture Notes in Computer Science
11. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-fourier lattice-based compact signatures over ntru (2019), https://api.semanticscholar.org/CorpusID:231637439
12. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing. p. 197–206. STOC '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1374376.1374407
13. Hanrot, G., Pujol, X., Stehlé, D.: Analyzing blockwise lattice algorithms using dynamical systems. In: Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference. Lecture Notes in Computer Science, vol. 6841, p. 441. Springer (2011). https://doi.org/10.1007/978-3-642-22792-9_25, https://www.iacr.org/archive/crypto2011/68410441/68410441.pdf
14. Kirshanova, E., Nguyen, H., Stehlé, D., Wallet, A.: On the smoothing parameter and last minimum of random orthogonal lattices. Des. Codes Cryptogr. **88**(5), 931–950 (2020). https://doi.org/10.1007/S10623-020-00719-W, https://doi.org/10.1007/s10623-020-00719-w
15. Klein, P.: Finding the closest lattice vector when it's unusually close. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms. p. 937–941. SODA '00, Society for Industrial and Applied Mathematics, USA (2000)
16. Li, J., Nguyen, P.Q.: A complete analysis of the BKZ lattice reduction algorithm. IACR Cryptol. ePrint Arch. p. 1237 (2020), https://eprint.iacr.org/2020/1237
17. Macbeath, A.M., Rogers, C.A.: Siegel's mean value theorem in the geometry of numbers. Mathematical Proceedings of the Cambridge Philosophical Society **54**(2), 139–151 (Apr 1958). https://doi.org/10.1017/S0305004100033302, https://www.cambridge.org/core/product/identifier/S0305004100033302/type/journal_article
18. Macbeath, A.M., Rogers, C.A.: Siegel's mean value theorem in the geometry of numbers. Mathematical Proceedings of the Cambridge Philosophical Society **54**(2), 139–151 (1958). https://doi.org/10.1017/S0305004100033302
19. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. In: 45th Annual IEEE Symposium on Foundations of Computer Science. pp. 372–381 (2004). https://doi.org/10.1109/FOCS.2004.72

20. Micciancio, D., Regev, O.: Lattice-based Cryptography, pp. 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_5
21. Pouly, A., Shen, Y.: Provable dual attacks on learning with errors. In: Advances in Cryptology – EUROCRYPT 2024: 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26–30, 2024, Proceedings, Part VII. p. 256–285. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-58754-2_10, https://doi.org/10.1007/978-3-031-58754-2_10
22. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (sep 2009). https://doi.org/10.1145/1568318.1568324, https://doi.org/10.1145/1568318.1568324
23. Rogers, C.A.: Mean values over the space of lattices. Acta Mathematica **94**(0), 249–287 (1955). https://doi.org/10.1007/BF02392493
24. Schnorr, C.P.: Lattice reduction by random sampling and birthday methods. In: Alt, H., Habib, M. (eds.) STACS 2003. pp. 145–156. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
25. Schnorr, C.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. **53**(2), 201–224 (jun 1987)
26. Siegel, C.L.: A Mean Value Theorem in Geometry of Numbers. The Annals of Mathematics **46**(2), 340 (Apr 1945). https://doi.org/10.2307/1969027, https://www.jstor.org/stable/1969027?origin=crossref
27. Stephens-Davidowitz, N.: On the Gaussian measure over lattices. Phd thesis, New York University (2017)
28. Wang, Z., Ling, C.: Lattice gaussian sampling by markov chain monte carlo: Bounded distance decoding and trapdoor sampling. IEEE Transactions on Information Theory **65**(6), 3630–3645 (2019). https://doi.org/10.1109/TIT.2019.2901497
29. Zamir, R., Nazer, B., Kochman, Y., Bistritz, I.: Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation and Multiuser Information Theory. Cambridge University Press (2014). https://doi.org/10.1017/CBO9781139045520