Discrete gaussian sampling for BKZ-reduced basis

Amaury Pouly[1] and Yixin Shen[2]

[1] Centre National de la Recherche Scientifique (CNRS), France
[2] Univ Rennes, Inria, CNRS, IRISA, Rennes, France

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 [PS24] 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 [GPV08, FHK+19], and in cryptanalysis as a fundamental tool for solving hard problems such as the Shortest Vector problem [ADRS15] or the Learning with Errors problem [PS24].

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 [MR04]. 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[1]) samplers [Kle00, GPV08, BLP+13, ACKS21] or very inefficient but arbitrarily good samplers [ADRS15]. 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 that the complexity smoothly interpolates between polynomial and exponential. Currently, the only[2] known sampler to do that is the Monte Carlo Markov Chain-based algorithm of [WL19],. It works for all values of s but the complexity formula is involved and depends significantly on 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 [WL19] when the basis follows a certain shape. This is the case for example when the basis is BKZ-reduced, a common occurrence in cryptanalysis.

E-mail: amaury.pouly@cnrs.fr (Amaury Pouly), yixin.shen@inria.fr (Yixin Shen)

[1] Above a quantity that is always strictly greater than the smoothing parameter.
[2] Although [BLP+13, Section 5] seems to imply a similar result, see Remark 1.

Date of this document: 2025-01-23.

In [PS24], the authors gave a simple approximation formula for the complexity of [WL19] 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, [PS24] 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 [WL19] for a BKZ-reduced basis. Our formula is valid over a wider range of values of s than [PS24] 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 [WL19] assuming the GSA. Furthermore, we conduct numerical experiments to compare the formula of [WL19] 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 [PS24] using our new formula, as well as other improvements in the code.

We also prove some results of independent interest on random q-ary 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 [WL19]. 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 [PS24]. 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

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

Recall that the Lambert W function is a multivalued function giving the complex solution(s) w to the equation $we^w = z$. In this paper we will only deal with real numbers. It can be shown that for any $x, y \in \mathbb{R}$, the equation

$$ye^y = x$$

can only be solved (for y) if $x \geqslant -\frac{1}{e}$. For positive numbers $x > 0$, this equation has exactly one real solution $y = W_0(x)$, where W_0 is one of the two real branches of the W function. It is known that W_0 is an increasing function.

We will use the following simple lemma on convex functions.

Lemma 1. *Let $a \leqslant b$ be integers and $f : \left[a-\frac{1}{2}, b+\frac{1}{2}\right] \to \mathbb{R}$ be a convex integrable function. Then $\sum_{i=a}^{b} f(i) \leqslant \int_{a-1/2}^{b+1/2} f(t)\,\mathrm{d}\,t$.*

Proof. We prove the result by induction on $b-a$. If $a=b$ then by Jensen inequality, we have that

$$f\left(\int_{a-\frac{1}{2}}^{a+\frac{1}{2}} t\,\mathrm{d}\,t\right) \leqslant \int_{a-\frac{1}{2}}^{a+\frac{1}{2}} f(t)\,\mathrm{d}\,t$$

which is exactly what we want since $\int_{a-\frac{1}{2}}^{a+\frac{1}{2}} t\,\mathrm{d}\,t = a$. The induction step is trivial by writing $\sum_{i=a}^{b} f(i) = f(a) + \sum_{i=a+1}^{b} f(i)$ and $\int_{a-1/2}^{b+1/2} f(t)\,\mathrm{d}\,t = \int_{a-1/2}^{a+1/2} f(t)\,\mathrm{d}\,t + \int_{a+1/2}^{b+1/2} f(t)\,\mathrm{d}\,t$, and applying the induction hypothesis twice. □

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 number. 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 $\mathrm{vol}(L_q(\mathbf{A})) = q^{n-\mathrm{rk}\,\mathbf{A}} \geqslant q^{n-k}$ and therefore $\mathrm{vol}(L_q^{\perp}(\mathbf{A})) = q^{\mathrm{rk}\,\mathbf{A}} \leqslant q^k$. 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 [ELZ05], [ZNKB14, Section 2.5.1], [MR09] or [PS24] 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 $s > 0$,

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

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. [Ste17] 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 [MR04] 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. 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 [Kle00, GPV08, BLP+13], including a time-space trade-off [ACKS21]. Sampling below the smoothing parameter is much more challenging and usually inefficient [ADRS15]. At the extreme, sampling for sufficiently small values of s allows one to solve the Shortest Vector problem (SVP) [ADRS15] which is known to be NP-hard under randomized reduction [Ajt98]. The Monte Carlo Markov Chain based algorithm of [WL19] 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 2. *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 $s > 0$,

$$0 < \rho_s(\mathbb{Z}) - \tilde{\rho}(s) \leqslant 2\sum_{k=2}^{\infty} e^{-\pi k^2} \leqslant \varepsilon := 6.974685811 \times 10^{-6}.$$

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 of $\rho_s(\mathbb{Z}) - \tilde{\rho}(s)$ is attained at $s = 1$. We can bound this value as follows:

$$\rho_1(\mathbb{Z}) - \tilde{\rho}(1) = 2e^{-4\pi} + 2\sum_{k=3}^{\infty} e^{-\pi k^2} \leqslant 2e^{-4\pi} + 2\sum_{k=9}^{\infty} e^{-\pi k} = 2e^{-4\pi} + \frac{2e^{-9\pi}}{1 - e^{-\pi}}$$

which is smaller than $6.974685811 \times 10^{-6}$ by numerical evaluation. □

2.3 The Metropolis-Hastings-Klein (MHK) algorithm

In [WL19], 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 [WL19] use the Klein algorithm [Kle00] 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 (in the number of steps[3]) to the stationary distribution. The main contribution

[3] More precisely, they show that the distance between the stationary distribution and the distribution after t steps is bounded by $(1 - \delta)^t$ where δ is the spectral gap.

of [WL19] 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 ([WL19, Theorem 1, (8), (23) and (24)[4]]). *There is an algorithm that given a basis of a lattice $L \subset \mathbb{R}^n$, 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 $\frac{1}{\Delta} = \frac{1}{\rho_s(\mathbf{t}+L)} \prod_{i=1}^{d} \rho_{s/\|\widetilde{\mathbf{b}}_i\|}(\mathbb{Z})$ and $\widetilde{\mathbf{b}}_1, \dots, \widetilde{\mathbf{b}}_d$ are the Gram-Schmidt vectors of the basis.*

Remark 1. It seems that a sampling algorithm with a similar complexity was previously described in [BLP+13]. More precisely, the statement [BLP+13, Lemma 2.3] only applies to large values of s, but the proof [BLP+13, Section 5] describes a rejection sampling algorithm that either outputs a sample exactly according to $D_{L,s}$, or outputs nothing with probability $1 - \Delta$ where $\Delta = \frac{\rho_s(\mathbf{t}+L)}{\prod_{i=1}^{d} \rho_{s/\|\widetilde{\mathbf{b}}_i\|}(\mathbb{Z})}$. Therefore if we run the algorithm until it outputs a sample, or output $\mathbf{t}$ after $N \in \mathbb{N}$ steps if we still did not get an output, we get a $N\mathsf{poly}(n)$ time algorithm. Call $\mathcal{D}_N$ the output distribution after N steps. Then it is not hard to see that $\mathrm{d_{TV}}(D_{L,s}, \mathcal{D}_N) = (1-\Delta)^N |1 - D_{L,s}(\mathbf{t})| \leqslant (1-\Delta)^N$. Therefore, for any $\varepsilon > 0$, $\mathrm{d_{TV}}(D_{L,s}, \mathcal{D}_N) \leqslant \varepsilon$ if $N \geqslant \frac{\ln \varepsilon}{\ln(1-\Delta)}$ which holds true if $N \geqslant \frac{\ln(1/\varepsilon)}{\Delta}$. It should be noted that the proof of [BLP+13, Section 5] requires that s be larger than a certain quantity (above the smoothing parameter) but this assumption only seems needed to prove the polynomial time complexity in [BLP+13, Lemma 2.3].

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 [PS24, 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 [PS24, Section 2.5] for more context.

Theorem 2 ([PS24, Theorem 3]). *Let $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$ and q be a prime number. Let $1 \leqslant p$ and $f : (\mathbb{Z}^n)^p \to \mathbb{R}$, then*

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

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

[4] [WL19] 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 number. For any $f : \mathbb{Z}^n \to \mathbb{R}$,*

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

Proof. Observe that by Theorem 2,

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

We now look at the various cases:

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

Therefore the only potentially non-zero terms are those for which $\mathrm{rk}_{\mathbb{Z}_q^n}(\mathbf{x},\mathbf{y}) = \mathrm{rk}_{\mathbb{Z}_q^n}(\mathbf{x}) = \mathrm{rk}_{\mathbb{Z}_q^n}(\mathbf{y}) = 1$. When this is the case, this means that there exists $\alpha, \beta \in \mathbb{Z}_q$ not both zero such that $\alpha\mathbf{x} + \beta\mathbf{y} = 0 \bmod q$. Furthermore, we must have $\beta \neq 0$ for otherwise we would have $\alpha\mathbf{x} = 0$ and therefore $\mathrm{rk}_{\mathbb{Z}_q^n}(\mathbf{x}) = 0$ which is not possible. Therefore by dividing by β, we can assume that $\beta = -1$. Therefore, $\mathbf{y} = \alpha\mathbf{x} \bmod q$, *i.e.* $\mathbf{y} = \alpha\mathbf{x} + q\mathbf{u}$ for some $\mathbf{u} \in \mathbb{Z}^n$. □

2.5 BKZ

The BKZ algorithm is a well-known lattice reduction algorithm [Sch87]. 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 [HPS11] or [LN20] to recent work on this topic.

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

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

By [HPS11], we have that $\delta_{\mathbf{B}}^d \leqslant 2\gamma_\beta^{\frac{d-1}{2(\beta-1)}+\frac{3}{2}}$ where γ_β is the β-Hermite constant. Experimentally, it has been verified [Che13] that

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

See [EJK20] 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) [Sch03] holds for any BKZ-β reduced basis.

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

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

The GSA is known to be reasonably accurate when $\beta \ll d$ and $\beta \geqslant 50$ which is the case in our experiments, but it does not correctly model what happens in the last $d - \beta$ coordinates. See [Bos21] 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 [WL19] primarily depends on the quantity

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

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 [WL19], 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 3. *For any $s > 0$, lattice L and $\widetilde{\mathbf{b}}_1, \dots, \widetilde{\mathbf{b}}_d$ the Gram-Schmidt vectors of a basis of L,*

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

Remark 2. When $\mathbf{t} \neq 0$ in (4), we cannot apply Lemma 3 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 [WL19, 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 [Reg09, Claim 3.8]. In this paper, we will only be interested in the case $\mathbf{t} = 0$.

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

$$\begin{aligned}
\frac{\prod_{i=1}^{d} \rho_{s/\|\widetilde{\mathbf{b}}_i\|}(\mathbb{Z})}{\rho_s(L)} &= \frac{\prod_{i=1}^{d} \frac{s}{\|\widetilde{\mathbf{b}}_i\|} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})}{\frac{s^d}{\operatorname{vol}(L)} \rho_{1/s}(\widehat{L})} \\
&= \frac{\operatorname{vol}(L)}{\prod_{i=1}^{d} \|\widetilde{\mathbf{b}}_i\|} \frac{\prod_{i=1}^{d} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})}{\rho_{1/s}(\widehat{L})} = \frac{\prod_{i=1}^{d} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})}{\rho_{1/s}(\widehat{L})}
\end{aligned}$$

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

This upper bound (the last inequality of Lemma 3) 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 3 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 [CDLP13] 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[5] Corollary 2, for any $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$, prime number 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. For cryptographic usage, we always take $k \ll n$, typically $k = n/2$ which that 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 and when $k \ll n$.

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 3:

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

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

4.1 How accurate is the GSA?

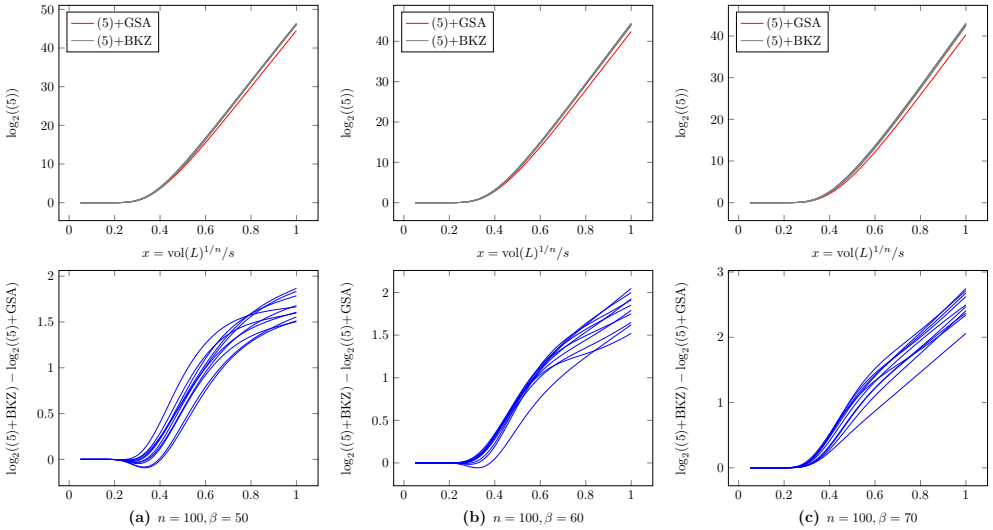
In this section, we compare the values given by (5) when using actual BKZ-reduced basis or when using the GSA (Heuristic 1) for the values of the $\|\widetilde{\mathbf{b}}_i\|$. We will refer to the former by "(5)+BKZ" and to the latter by "(5)+GSA".

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 $d = n$ (i.e. full-rank lattices) and β, we pick $N = 10$ bases at random and BKZ-β reduce them. For BKZ, we use the G6K software from [ADH+19]. Specifically, we use the "pump-and-jump" strategy with $\frac{n^2}{\beta^2}\log(n)$ tours (see [LN24] for a theoretical argument).

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

[5] Proven later in Section 6 which is independent from the rest of the paper.



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

Figure 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.

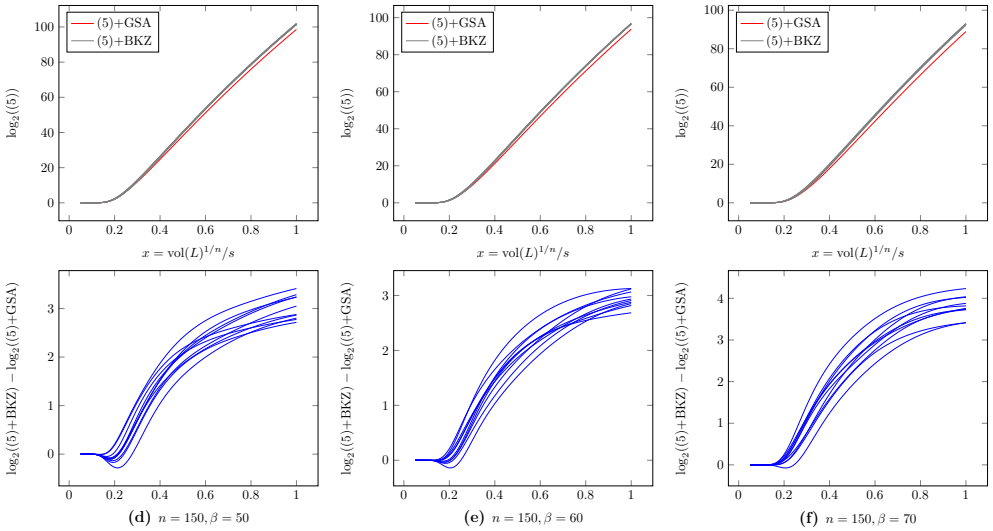
$x = \mathrm{vol}(L)^{1/n}/s$ (for fixed n and β), we plot all curves as a function of x. As discussed in Section 4, the upper bound (5) is only tight for values of s that satisfy $s \gtrapprox \mathrm{vol}(L)^{1/k}$, *i.e.* $x \leqslant 1$. Therefore we only plot the curves over the interval $[0, 1]$. 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 [Bos21, DEP23] that running BKZ on the standard[6] 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 observed that the GSA seems to give better results than the Z-shape adapted GSA, which is why in Figure 3 we plot (5)+GSA. We leave as an open question to explain why this is the case.

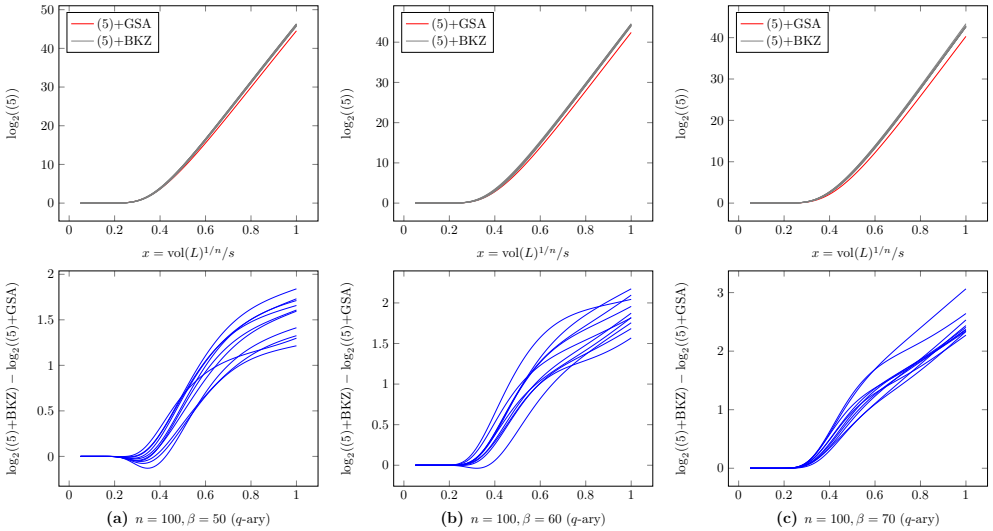
The results can be found in Figure 1, Figure 2 and Figure 3. We observe a reasonably good agreement between (5)+BKZ and (5)+GSA. Unsurprisingly, the error increases as s becomes smaller (and x becomes closer to 1) but we expect that most applications of this result will only use small values of x. In 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 $1/4$.

[6] 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 $\mathbf{B}$.



(d) $n = 150, \beta = 50$ (e) $n = 150, \beta = 60$ (f) $n = 150, \beta = 70$

Figure 2: 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)

Figure 3: 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.

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 [PS24]), the cost of evaluating this formula can quickly become prohibitive. Indeed, evaluating (5) takes time $O(n)$ to evaluate, compared to $O(1)$ to the formula that we give.

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

$$\ln\left(\prod_{i=1}^{d} \rho_{\|\widetilde{\mathbf{b}}_i\|/s}(\mathbb{Z})\right) \leqslant A + \sum_{\ell=1}^{p} \frac{2^\ell(-1)^{\ell+1}}{\ell}(B_\ell + C_\ell)$$

where

$$A = (d_0+1)\ln\frac{\alpha}{H_\beta^{d_0}} + d\varepsilon,$$

$$B_\ell = \frac{E_1\left(\pi\ell\alpha^2 H_\beta^{-4(d_0+\frac{1}{2})}\right) - E_1(\pi\ell\alpha^2 H_\beta^{-2})}{2\ln(H_\beta)},$$

$$C_\ell = \frac{E_1\left(\frac{\pi}{\alpha^2}\ell H_\beta^{4(d-\frac{1}{2})}\right) - E_1\left(\frac{\pi}{\alpha^2}\ell H_\beta^{4(d_0+\frac{1}{2})}\right)}{2\ln(H_\beta)},$$

$d_0 = \max\left(-1, \min\left(d-1, \left\lfloor \frac{\ln(\alpha)}{2\ln(H_\beta)} \right\rfloor\right)\right)$ *and ε comes from Lemma 2.*

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

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

Now check that

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

We let

$$d_0 = \max\left(-1, \min\left(d-1, \left\lfloor \frac{\ln(\alpha)}{2\ln(H_\beta)} \right\rfloor\right)\right)$$

so that

$$\sum_{i=0}^{d-1} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) = \sum_{i=0}^{d_0} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) + \sum_{i=d_0+1}^{d-1} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}).$$

For $0 \leqslant i \leqslant d_0$, we have $\alpha H_\beta^{-2i} \geqslant 1$ by definition of d_0. Then by Lemma 2, there exists ε such that $\rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) \leqslant \alpha H_\beta^{-2i} \cdot \left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right) + \varepsilon$. By using that $\ln(x+\varepsilon) \leqslant \ln(x) + \varepsilon$ for any $x \geqslant 1$, we get that

$$\sum_{i=0}^{d_0} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z})$$

$$\begin{aligned}
&\leqslant \sum_{i=0}^{d_0} \ln\left(\alpha H_\beta^{-2i} \cdot \left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right)\right) + \sum_{i=0}^{d_0} \varepsilon \\
&= \sum_{i=0}^{d_0} \ln\left(\alpha H_\beta^{-2i}\right) + \sum_{i=0}^{d_0} \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right) + (d_0+1)\varepsilon \\
&= (d_0+1)\left(\varepsilon + \ln\frac{\alpha}{H_\beta^{d_0}}\right) + \sum_{i=0}^{d_0} \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right)
\end{aligned}$$

by a routine calculation. Now $i \mapsto \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right)$ is a convex function over the interval $[-\frac{1}{2}, d_0 + \frac{1}{2}]$ whenever $H_\beta \leqslant \frac{\sqrt{\pi}}{\sqrt{1+W_0(\frac{2}{e})}}$ (see Section A.1). Therefore by Lemma 1,

$$\sum_{i=0}^{d_0} \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right) \leqslant \int_{-\frac{1}{2}}^{d_0+\frac{1}{2}} \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4t})\right) \mathrm{d}\, t.$$

Recall that for odd p, $\ln(1+x) \leqslant \sum_{\ell=1}^{p} \frac{(-1)^{p+1}x^p}{p}$ for any $x \geqslant 0$. It follows that

$$\begin{aligned}
\sum_{i=0}^{d_0} \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right) &\leqslant \int_{-\frac{1}{2}}^{d_0+\frac{1}{2}} \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right) \\
&\leqslant \int_{-\frac{1}{2}}^{d_0+\frac{1}{2}} \sum_{\ell=1}^{p} \frac{2^\ell(-1)^{\ell+1}}{\ell} \exp\left(-\pi\ell\alpha^2 H_\beta^{-4t}\right) \mathrm{d}\, t \\
&= \sum_{\ell=1}^{p} \frac{2^\ell(-1)^{\ell+1}}{\ell} \int_{-\frac{1}{2}}^{d_0+\frac{1}{2}} \exp\left(-\pi\ell\alpha^2 H_\beta^{-4t}\right) \mathrm{d}\, t.
\end{aligned}$$

For any $x \neq 1$,

$$\begin{aligned}
\int_a^b \exp(-yx^{4t}) \,\mathrm{d}\, t &= \int_{x^{4a}}^{x^{4b}} \frac{\exp(-yu)}{4\ln(x)u} \,\mathrm{d}\, u && \text{by the change } u = x^{4t} \\
&= \frac{E_1(yx^{4a}) - E_1(yx^{4b})}{4\ln(x)} && \text{by (2).}
\end{aligned} \tag{6}$$

Therefore,

$$\begin{aligned}
\sum_{i=0}^{d_0} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) \leqslant (d_0+1)&\left(\varepsilon + \ln\frac{\alpha}{H_\beta^{d_0}}\right) \\
&+ \sum_{\ell=1}^{p} \frac{2^\ell(-1)^{\ell+1}}{\ell} \frac{E_1\left(\pi\ell\alpha^2 H_\beta^{-4(d_0+\frac{1}{2})}\right) - E_1(\pi\ell\alpha^2 H_\beta^{-2})}{2\ln(H_\beta)}.
\end{aligned}$$

Similarly, for $d_0 < i \leqslant d$, we have $\alpha H_\beta^{-2i} \leqslant 1$ so $\rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) \leqslant 1 + 2\exp(-\frac{\pi}{\alpha^2} H_\beta^{4i})$ by Lemma 2. It follows by the same argument as above that

$$\sum_{i=d_0+1}^{d-1} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) \leqslant \sum_{i=d_0+1}^{d-1} \ln\left(1 + 2\exp(-\tfrac{\pi}{\alpha^2} H_\beta^{4i})\right) + (d-1-d_0)\varepsilon.$$

Now $i \mapsto \ln\left(1+2\exp(-\frac{\pi}{\alpha^2}H_\beta^{4i})\right)$ is a convex function over the interval $[d_0+\frac{1}{2}, d-\frac{1}{2}]$ whenever $H_\beta \leqslant \frac{\sqrt{\pi}}{\sqrt{1+W_0(\frac{2}{e})}}$ (see Section A.1). Therefore by Lemma 1,

$$\sum_{i=0}^{d_0} \ln\left(1+2\exp(-\tfrac{\pi}{\alpha^2}H_\beta^{4i})\right) \leqslant \int_{d_0+\frac{1}{2}}^{d-\frac{1}{2}} \ln\left(1+2\exp(-\tfrac{\pi}{\alpha^2}H_\beta^{4t})\right)\,\mathrm{d}\,t.$$

It then follows by the same argument as above that

$$\sum_{i=d_0+1}^{d-1} \ln \rho_{\alpha H_\beta^{-2i}}(\mathbb{Z}) \leqslant (d-1-d_0)\varepsilon + \sum_{\ell=1}^{p} \frac{2^\ell(-1)^{\ell+1}}{\ell} \frac{E_1\left(\frac{\pi}{\alpha^2}\ell H_\beta^{4(d-\frac{1}{2})}\right) - E_1\left(\frac{\pi}{\alpha^2}\ell H_\beta^{4(d_0+\frac{1}{2})}\right)}{2\ln(H_\beta)}.$$

□

4.3 How accurate is the approximation?

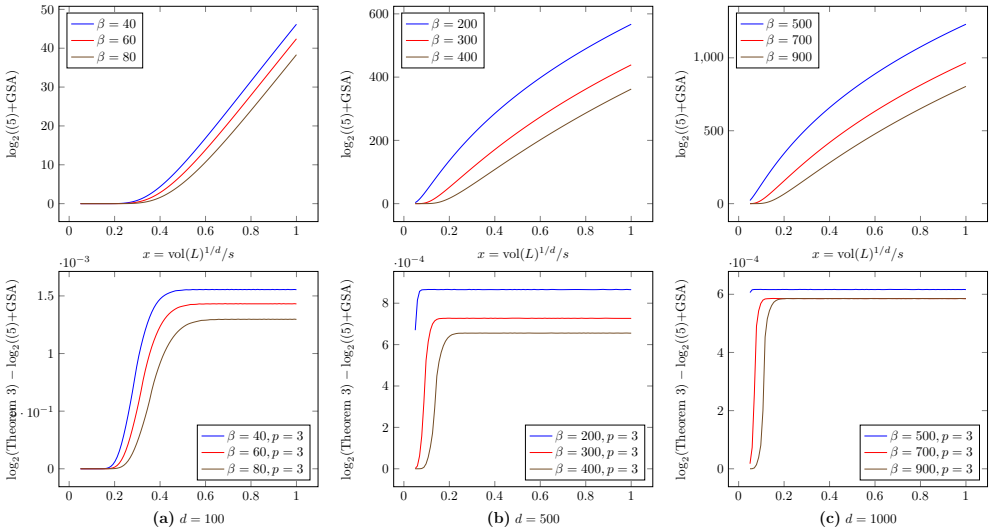
We now compare the formula of Theorem 3 with the upper bound (5) on the complexity where we use the GSA (Heuristic 1) for the values of the $\|\tilde{\mathbf{b}}_i\|$. We will refer to the latter by "(5)+GSA" as we did in Section 4.1.

We observe that both (5)+GSA and the formula from Theorem 3 only depend on d, $x = \mathrm{vol}(L)^{1/d}/s$ 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 $s \gtrapprox \mathrm{vol}(L)^{1/d}$, *i.e.* $x \leqslant 1$. Therefore we only plot the curves over the interval $[0,1]$.

The curves can be found in Figure 4, we used $p=3$ in Theorem 3 for all curves. The bottom figures confirm that the difference between Theorem 3 and (5)+GSA is negligible. Indeed, we can see that for $d=1000$, the logarithm of the ratio between the two quantities is less than 0.001, meaning that the approximation is correct within a multiplicative factor $2^{0.001} \leqslant 1.00067$. This factor should be negligible for virtually all applications given that the complexities grows exponentially in d, as can be seen on the top figures. Although not shown here, we observed that the difference between (5)+GSA and the formula from Theorem 3 is much larger for $p=1$ compared to $p=3$ where for $d=1000$ it is approximately 0.15 instead of $6\cdot 10^{-4}$.

5 Applications to dual attack on LWE

In this section, we revisit the complexity estimates from [PS24] using our approximation formula (Theorem 3). The approach in [PS24] 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.



(a) $d = 100$ (b) $d = 500$ (c) $d = 1000$

Figure 4: Top pictures: (logarithm) of the complexity upper bound given by (5)+GSA, for different values of d 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.

5.1 High-level overview of the attack

In this section, we give a succint presentation of the attack in [PS24]. 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 [PS24, 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 $\mathbf{w}_1, \dots, \mathbf{w}_N$ 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 [PS24] uses the sampler from [WL19], 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[7] to use our new approximate formula. This requires a few more changes since the optimizer of [PS24] enforces the condition[8] that $s \geqslant \frac{\|\mathbf{b}_1\|}{2q}$ [PS24, Section 4.4]. Furthermore, the optimizer of [PS24] 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[9] 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 to 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 [PS24], $d = m$, $\text{vol}(L) = q^{n_{\text{dual}}}$ and $t = qs$. Note that similarly to [PS24], 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). Importantly, all estimates in the table **ignore polynomial factors**, including that of Theorem 1.

We observe some significant improvements in the complexity compared to [PS24], especially without modulus switching, thanks to the smaller values of s that our formula

[7] The code for the complexity estimates in [PS24] is available as an artifact.

[8] Beware that the algorithm of [PS24] actually samples at qs and not s.

[9] By the Gaussian heuristic, which essentially holds true for random q-ary lattices [PS24, 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.

Table 1: Dual attack cost estimates and their parameters as described in [PS24, 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 [PS24] 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 [PS24]
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

Table 2: Dual attack cost estimates and their parameters as described in [PS24, 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 [PS24] 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 [PS24]
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

is able to handle. However, when looking in detail at the results, we also observe that the optimizer of [PS24] 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 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 extrapolate 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.

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. A similar result was shown for "standard" random q-ary lattices (i.i.d. from uniform entries) in [LLBS14, Lemma 3] but only gives the expected value, whereas we also bound the variance. A closely related result is available in [KNSW20] which studies matrices with each entry independently and identically distributed from an integer Gaussian distribution. Similarly, [CPS+20, Appendix A], [LPR13, Section 7] and [SS11, Theorem 2] analyzes the Gaussian mass of a random q-ary lattice over cyclotomic fields.

Lemma 4. *For any $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$, prime number q and $s > 0$,*

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

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

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

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

Proof. Let ε be as in Lemma 2. Clearly $s \geqslant 1$ if $s = \xi q^{k/n}$ so we can apply Lemma 2 to get that

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

By Lemma 2, 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 number q, $\xi > 1$ and α, if $s = \xi q^{k/n} \leqslant q/2$, $q^{k/n} \geqslant 2$ 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. Let f be defined as in Lemma 5 which is a decreasing function. Observe that if $2s \leqslant q$ then $2\xi q^{k/n} \leqslant q$, 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 $\varepsilon \leqslant 6.98 \times 10^{-6}$ we have $(1+\varepsilon)f(2) \leqslant 1.000014$. Hence, by Lemma 4 and Lemma 5

$$\begin{aligned} \mu &:= \mathbb{E}_{L \sim \mathcal{L}^{\perp}_{n,k,q}}\left[\rho_{1/s}(\widehat{L})\right] \\ &\leqslant \rho_{1/s}(\mathbb{Z}^n) + q^{k-n}\rho_{q/s}(\mathbb{Z}^n), \\ &\leqslant (1+\varepsilon)^n f(s)^n + (1+\varepsilon)^n \xi^{-n} f\left(q^{1-k/n}/\xi\right)^n \\ &\leqslant (1+\varepsilon)^n f(2)^n + (1+\varepsilon)^n \xi^{-n} f\left(2\right)^n \\ &\leqslant 1.000007^n + \xi^{-n} \cdot 1.000014^n, \end{aligned}$$

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.

Acknowledgements

We thank Léo Ducas for pointing out to us that [BLP+13, Section 5] also contains a DGS sampler of similar complexity to that of [WL19].

References

[ACKS21] Divesh Aggarwal, Yanlin Chen, Rajendra Kumar, and Yixin Shen. Improved (provable) algorithms for the shortest vector problem via bounded distance decoding. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 4:1–4:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.4.

[ADH+19] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 717–746, Cham, 2019. Springer International Publishing.

[ADRS15] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time using discrete gaussian sampling: Extended abstract. In Rocco A. Servedio and Ronitt 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 733–742. ACM, 2015. doi:10.1145/2746539.2746606.

[Ajt98] Miklós Ajtai. 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*, STOC '98, page 10–19, New York, NY, USA, 1998. Association for Computing Machinery. doi:10.1145/276698.276705.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 575–584, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2488608.2488680.

[Bos21] *Lattice Attacks on NTRU and LWE: A History of Refinements*, page 15–40. London Mathematical Society Lecture Note Series. Cambridge University Press, 2021.

[CDLP13] Kai-Min Chung, Daniel Dadush, Feng-Hao Liu, and Chris Peikert. On the lattice smoothing parameter problem. In *2013 IEEE Conference on Computational Complexity*, pages 230–241, 2013. doi:10.1109/CCC.2013.31.

[Che13] Yuanmi Chen. *Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe*. PhD thesis, 2013. Thèse de doctorat dirigée par Nguyen, Phong Q. Informatique Paris 7 2013. URL: http://www.theses.fr/2013PA077242.

[CPS+20] Chitchanok Chuengsatiansup, Thomas Prest, Damien Stehlé, Alexandre Wallet, and Keita Xagawa. Modfalcon: Compact signatures based on module-ntru

lattices. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '20, page 853–866, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3320269.3384758.

[DEP23] Léo Ducas, Thomas Espitau, and Eamonn W. Postlethwaite. Finding short integer solutions when the modulus is small. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 150–176, Cham, 2023. Springer Nature Switzerland.

[EJK20] Thomas Espitau, Antoine Joux, and Natalia Kharchenko. 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 Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology – INDOCRYPT 2020*, volume 12578, page 440–462. Springer International Publishing, Cham, 2020. Series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-030-65277-7_20.

[ELZ05] U. Erez, S. Litsyn, and R. Zamir. Lattices which are good for (almost) everything. *IEEE Transactions on Information Theory*, 51(10):3401–3416, 2005. doi:10.1109/TIT.2005.855591.

[FHK+19] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. 2019. URL: https://api.semanticscholar.org/CorpusID:231637439.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 197–206, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1374376.1374407.

[HPS11] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference*, volume 6841 of *Lecture Notes in Computer Science*, page 441. Springer, 2011. URL: https://www.iacr.org/archive/crypto2011/68410441/68410441.pdf, doi:10.1007/978-3-642-22792-9_25.

[Kle00] Philip Klein. Finding the closest lattice vector when it's unusually close. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, page 937–941, USA, 2000. Society for Industrial and Applied Mathematics.

[KNSW20] Elena Kirshanova, Huyen Nguyen, Damien Stehlé, and Alexandre Wallet. On the smoothing parameter and last minimum of random orthogonal lattices. *Des. Codes Cryptogr.*, 88(5):931–950, 2020. URL: https://doi.org/10.1007/s10623-020-00719-w, doi:10.1007/S10623-020-00719-W.

[LLBS14] Cong Ling, Laura Luzzi, Jean-Claude Belfiore, and Damien Stehlé. Semantically secure lattice codes for the gaussian wiretap channel. *IEEE Transactions on Information Theory*, 60(10):6399–6416, 2014. doi:10.1109/TIT.2014.2343226.

[LN20] Jianwei Li and Phong Q. Nguyen. A complete analysis of the BKZ lattice reduction algorithm. *IACR Cryptol. ePrint Arch.*, page 1237, 2020. URL: https://eprint.iacr.org/2020/1237.

[LN24] Jianwei Li and Phong Q. Nguyen. A complete analysis of the bkz lattice reduction algorithm. *J. Cryptol.*, 38(1), December 2024. doi:10.1007/s00145-024-09527-0.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 35–54, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 372–381, 2004. doi:10.1109/FOCS.2004.72.

[MR09] Daniele Micciancio and Oded Regev. *Lattice-based Cryptography*, pages 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-540-88702-7_5.

[PS24] Amaury Pouly and Yixin Shen. 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*, page 256–285, Berlin, Heidelberg, 2024. Springer-Verlag. doi:10.1007/978-3-031-58754-2_10.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), sep 2009. doi:10.1145/1568318.1568324.

[Sch87] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53(2):201–224, jun 1987.

[Sch03] Claus Peter Schnorr. Lattice reduction by random sampling and birthday methods. In Helmut Alt and Michel Habib, editors, *STACS 2003*, pages 145–156, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[SS11] Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 27–47, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Ste17] Noah Stephens-Davidowitz. *On the Gaussian measure over lattices*. Phd thesis, New York University, 2017.

[WL19] Zheng Wang and Cong Ling. Lattice gaussian sampling by markov chain monte carlo: Bounded distance decoding and trapdoor sampling. *IEEE Transactions on Information Theory*, 65(6):3630–3645, 2019. doi:10.1109/TIT.2019.2901497.

[ZNKB14] Ram Zamir, Bobak Nazer, Yuval Kochman, and Ilai Bistritz. *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation and Multiuser Information Theory*. Cambridge University Press, 2014. doi:10.1017/CBO9781139045520.

Appendix

A Deferred results

A.1 Convexity of a certain function

First, we claim the following result.

Lemma 6. *Let $a > 0$, $x \neq 1$ and $f(t) = \ln\left(1 + 2\exp(-ax^t)\right)$ for any $t \in \mathbb{R}$. Let $t_0 = \frac{1}{\ln x} \ln\left(\frac{1+W\left(\frac{2}{e}\right)}{a}\right)$. If $0 < x < 1$ then f is a convex function over $(-\infty, t_0]$. If $x > 1$ then f is a convex function over $[t_0, \infty)$.*

Proof. Clearly f is twice differentiable and a routine calculation shows that

$$f''(t) = \frac{2ax^t \ln(x)^2 \exp(-ax^t)}{\left(1 + 2\exp(-ax^t)\right)^2} \underbrace{\left(ax^t - 1 - 2\exp(-ax^t)\right)}_{:=g(t)}.$$

Now observe that

$$\begin{aligned} g(t) = 0 &\Leftrightarrow ax^t - 1 = 2\exp(-ax^t) \\ &\Leftrightarrow ax^t - 1 = \tfrac{2}{e}\exp(1 - ax^t) \\ &\Leftrightarrow ax^t - 1 = W_0(-\tfrac{2}{e}) \\ &\Leftrightarrow x^t = \frac{1 + W_0(\frac{2}{e})}{a} \\ &\Leftrightarrow t = \frac{1}{\ln x} \ln\left(\frac{1 + W_0(\frac{2}{e})}{a}\right) \end{aligned}$$

where W denotes the Lambert W function. If $x < 1$ then $g(t) \to \infty$ as $t \to -\infty$ so g and therefore f'' is positive over the interval $(-\infty, t_0)$ where t_0 is the unique solution to $g(t_0) = 0$ above. If $x > 1$ then $g(t) \to \infty$ as $t \to \infty$ so g and therefore f'' is positive over the interval $[t_0, \infty)$. □

Recall the setting from the proof of Theorem 3: we have $\alpha > 0$, $H_\beta > 1$ and $d_0 = \max\left(-1, \min\left(k - 1, \left\lfloor \frac{\ln(\alpha)}{2\ln(H_\beta)} \right\rfloor\right)\right)$. We want to show that $i \mapsto \ln\left(1 + 2\exp(-\pi\alpha^2 H_\beta^{-4i})\right)$ is a convex function over the interval $[-\frac{1}{2}, d_0 + \frac{1}{2}]$. First if $d_0 = -1$ then the result is trivial. Otherwise, we can assume that $d_0 \geqslant 0$ and therefore $d_0 \leqslant \left\lfloor \frac{\ln(\alpha)}{2\ln(H_\beta)} \right\rfloor \leqslant \frac{\ln(\alpha)}{2\ln(H_\beta)}$. Let $a = \pi\alpha^2$ and $x = H_\beta^{-4} \in (0, 1)$. By Lemma 6, this function is convex over the interval $(-\infty, t_0)$ where

$$t_0 = \frac{1}{\ln x} \ln\left(\frac{1 + W\left(\frac{2}{e}\right)}{a}\right) = \frac{1}{-4\ln H_\beta} \ln\left(\frac{1 + W\left(\frac{2}{e}\right)}{\pi\alpha^2}\right) = \frac{\ln \alpha}{2\ln H_\beta} + \frac{\ln \frac{1+W\left(\frac{2}{e}\right)}{\pi}}{-4\ln H_\beta}.$$

Since $\frac{\ln \alpha}{2\ln H_\beta} \geqslant d_0$, we have $t_0 \geqslant d_0 + \frac{1}{2}$ whenever

$$\frac{\ln \frac{1+W\left(\frac{2}{e}\right)}{\pi}}{-4\ln H_\beta} \geqslant \tfrac{1}{2} \quad \Leftrightarrow \quad H_\beta \leqslant \frac{\sqrt{\pi}}{\sqrt{1 + W_0(\frac{2}{e})}}.$$

However one can verify that this last inequality always holds because the right-hand side is approximately 1.4653 whereas the H_β is always smaller than $H_{36} \approx 1.012608$, as can be verified by numerical computations. Similarly, we want to show that $i \mapsto \ln\left(1 + 2\exp(-\frac{\pi}{\alpha^2} H_\beta^{4i})\right)$ is a convex function over the interval $[d_0 + \frac{1}{2}, k - \frac{1}{2}]$. First if $d_0 = k-1$ then the result is trivial. Otherwise we can assume that $d_0 < k-1$ and therefore that $d_0 \geqslant \left\lfloor \frac{\ln(\alpha)}{2\ln(H_\beta)} \right\rfloor \geqslant \frac{\ln(\alpha)}{2\ln(H_\beta)} - 1$. Let $a = \frac{\pi}{\alpha^2}$ and $x = H_\beta^4 > 1$. By Lemma 6, this function is convex over the interval $[t_1, \infty)$ where

$$t_1 = \frac{1}{\ln x} \ln\left(\frac{1 + W\left(\frac{2}{e}\right)}{a}\right) = \frac{1}{4\ln H_\beta} \ln\left(\frac{1 + W\left(\frac{2}{e}\right)}{\pi} \alpha^2\right) = \frac{\ln \alpha}{2\ln H_\beta} + \frac{\ln \frac{1+W\left(\frac{2}{e}\right)}{\pi}}{4\ln H_\beta}.$$

Since $\frac{\ln \alpha}{2\ln H_\beta} \leqslant d_0 + 1$, we have $t_1 \leqslant d_0 - \frac{1}{2}$ whenever

$$\frac{\ln \frac{1+W\left(\frac{2}{e}\right)}{\pi}}{4\ln H_\beta} \leqslant -\frac{1}{2} \quad \Leftrightarrow \quad H_\beta \leqslant \frac{\sqrt{\pi}}{\sqrt{1 + W_0(\frac{2}{e})}}.$$