Refined Modelling of the Primal Attack, and Variants Against Module-LWE

Paola de Perthuis[1] and Filip Trenkić[2]

[1] Centrum Wiskunde & Informatica (CWI), Amsterdam, the Netherlands
[2] King's College London, United Kingdom

Abstract The primal attack reduces Learning with Errors (LWE) to the unique Shortest Vector Problem (uSVP), and then applies lattice reduction such as BKZ to solve the latter. Estimating the cost of the attack is required to evaluate the security of constructions based on LWE.
Existing fine-grained estimators for the cost of the primal attack, due to Dachman-Soled–Ducas–Gong–Rossi (CRYPTO 2020) and Postlethwaite–Virdia (PKC 2021), differ from experimental data as they implicitly assume the unique shortest vector is resampled several times during the attack, changing its length. Furthermore, these estimators consider only the first two moments of the LWE secret and error, and therefore do not differentiate between distinct centred distributions with equal variances. We remedy both issues by initially fixing the short vector's length, and later integrating over its distribution. We provide extensive experimental evidence that our estimators are more accurate and faithfully capture the behaviour of different LWE distributions.
In the case of Module-LWE, lattice reduction utilising the module structure could lead to cheaper attacks. We build upon the analysis of module lattice reduction by Ducas–Engelberts–Perthuis (Asiacrypt 2025), providing a simulator for Module-BKZ generalising the BKZ simulator of Chen–Nguyen (Asiacrypt 2011). We design estimators for a module variant of the primal attack, supporting our analysis with experimental evidence. Asymptotically, we show the module primal attack over a degree d number field K has a reduced cost, resulting in a subexponential gain, whenever the discriminant Δ_K satisfies $|\Delta_K| < d^d$, one such case being non-power-two cyclotomics.

Keywords: Cryptanalysis · Module Lattices · Learning With Errors

1 Introduction

The Learning with Errors (LWE) and NTRU problems serve as central hardness assumptions in lattice-based cryptography, underpinning many primitives and advanced constructions. Module-LWE (MLWE), the algebraically structured variant of LWE, is used extensively in post-quantum standards including CRYSTALS-Kyber [BDK+18] and SMAUG-T [CCHY24], while NTRU+ [KP22] relies on the hardness of NTRU.

The *primal attack* refers to a family of attacks which reduce LWE and NTRU to the unique Shortest Vector Problem (uSVP), then apply lattice reduction algorithms, the most common of which is BKZ [Sch87, SE94]. The BKZ algorithm is parameterised by a blocksize β, which corresponds to the quality of lattice reduction, and implemented variants have complexity $2^{\Theta(\beta)}$ [LN20]. Evaluating the cost of the primal attack is thus directly dependent on estimating the minimal blocksize β such that BKZ-β recovers the unique shortest vector [APS15, ADPS16, ACD+18].

Initial estimates gave a single blocksize β which was likely to succeed in solving the instance [ADPS16], but later works reported that smaller blocksizes often have a non-negligible probability of success [AGVW17, BMW19]. More recent works [DDGR20, PV21] aimed to capture these low-probability events by instead estimating, for each blocksize β, the probability that BKZ-β would recover the unique shortest vector. These models rely on

a BKZ simulator [CN11, BSW18], which predicts the evolution of the profile – the lengths of vectors in the lattice basis – during the execution of BKZ. We note that these estimators are designed for LWE instances where the secret and error are sampled coordinatewise from a discrete Gaussian distribution, but in practice a variety of other distributions are used. One unsatisfying solution provided in [PV21] is to model an arbitrary distribution by a discrete Gaussian of the same variance.

Many real-world schemes use structured lattice assumptions, such as NTRU and MLWE, where the lattices arise as modules over the ring of integers in a number field [HPS98, LPR10, LS15]. The use of module lattices can be crucial for efficiency and small key sizes, but the algebraic structure opens potential avenues for attack. In [LPSW19, MS20], module variants of the standard reduction algorithms were first proposed, with a study of their behavior on worst-case instances. Analysis of experimental behavior was not immediate, because of the challenges posed by the implementation of module-lattice reduction. Then, [KK24] gave a predictive analysis of the reduction for NTRU instances, with a negative result. Most recently, [DEP25] provided an average-case study of the Module-BKZ (mBKZ) algorithm, predicting the quality of mBKZ-reduced lattice bases, depending on the blocksize and the underlying number field.

Crucially, it was shown that for lattices over non-power-two cyclotomic number fields, the Module-BKZ algorithm achieves the same basis quality as BKZ-β while using a sublinearly smaller blocksize, in particular providing a gain in $\Theta(\beta/\ln\beta)$.

Contributions. Broadly, this article seeks the comprehension of dependencies in the primal attack. While a unique short vector may be initially drawn from a random distribution, it does not change during the attack itself; and when considering attacks against module lattices, there are multiple short vectors resulting from the action of roots of unity, which do not behave independently. Our goal is to introduce statistical models which account for and quantify these phenomena.

Our first contribution is to refine existing estimators [DDGR20, PV21] for the success probability of the primal attack. For small blocksizes where experiments are feasible, these lack accuracy, and we explain this is due to a heuristic that implicitly assumes the short vector is resampled several times during the attack (§3.1). We provide new estimators to remedy this. Our technique is to model the success probability assuming the length λ_1 of the shortest vector is known, and then integrate over the randomness of λ_1 (§3.2). A similar technique was used in [DP26] to improve analysis of the dual attack. Concretely, we use that the projection of the short vector into a random subspace follows a beta distribution, whereas previous works use a chi-squared distribution – this discrepancy was noted in [BN24]. We provide extensive experimental evidence, concluding that our estimators are more accurate (Figure 3) and moreover able to differentiate between LWE distributions with the same mean and variance (Figure 5).

Our second contribution is to build upon the analysis of Module-BKZ in [DEP25], in particular addressing Open Questions 1, 2 and 4. First, we propose a simulator analogous to the BKZ simulator of [CN11] for the evolution of a basis profile during Module-BKZ (§4). Using this, we adapt our estimators to predict the success probability of a module analogue of the primal attack, which can outperform the standard primal attack. We support our analysis with experiments (§5.4), though because of the limitations of current implementations, they were conducted for cyclotomic number fields of low degree. We hence complement them with an asymptotic analysis (§5.2). In particular, we show that if β is the $\mathbb{Q}$-blocksize required by Module-BKZ to solve Module-LWE, then BKZ requires a blocksize of

$$\beta_{\mathrm{eq}} = \beta + \frac{1}{d}\ln\frac{d^d}{|\Delta_K|}\ln\left(\frac{q^u|\Delta_K|^{1/d}}{\sigma^2 2\pi e}\right)\frac{\beta}{(\ln\beta)^2} + o\left(\frac{\beta}{(\ln\beta)^2}\right).$$

where K is the underlying degree-d number field with discriminant Δ_K, q and σ are the MLWE modulus and noise width, and $u \in (0, 2)$ pertains to the number of samples. Hence,

the sublinear gain provided by Module-BKZ for non-power-two cyclotomics translates to a sublinear gain on the blocksize for which the module primal attack succeeds.

Finally, for this comparison to be exhaustive, we analyse variants of the standard primal attack against MLWE (§6). Rather than reducing to unique-SVP, one can embed several short vectors in the module lattice, obtained from the action of the roots of unity in the number field: then, finding any one of them recovers the secret. Such attacks were proposed in [NY21, UOM24], with experimental evidence to suggest embedding more short vectors can be beneficial, which we confirm by proposing an extension of our estimators to this scenario, and comparing to experiments.

Our code is open source and available at `https://github.com/FTcode/beta-is-better` (§3) and `https://github.com/FTcode/mPrimal` (§4, §5, §6).

Cryptanalytic Impact. This work entails the following consequences for the cryptographic security of lattice-based schemes.

1. Better security estimates will be available for Learning With Errors (LWE)-based schemes, using our refined modelling of the distribution of the norm of the shortest vector during the primal attack. In particular, we show that previous estimates generally underestimated the security by expecting slightly smaller blocksizes than those provided by experiments and our estimations.
2. Furthermore, we allow security estimates tailored to arbitrary LWE secret and error distributions, superseding the existing paradigm where all distributions are approximated as Gaussians.
3. Our precise simulations for module-lattice reduction yield new security estimates for MLWE-based schemes, by taking into account the improved performance of a module primal attack exploiting the algebraic structure. In particular, we exhibit a subexponential gain when using non-power-of-two cyclotomics or more generally number fields with a small discriminant.
4. For power-of-two cyclotomic fields, our simulations show faster short vector recovery when using Module-BKZ, making use of more oracle calls in the terminal block. This mitigates the negative result of [DEP25], as a faster convergence to a worse profile may improve cryptanalysis in practice when the number of reduction tours is limited.
5. Our asymptotic analysis comparing the performance of the standard primal attack and the module primal attack on an MLWE instance shows that the improved slope of Module-BKZ translates to a smaller required blocksize when attacking MLWE for non-power-of-two cyclotomic number fields.

We note that real-world cryptographic constructions typically use low-rank instances over number fields with a large degree d. In this case, even the smallest Module-BKZ $\mathbb{Q}$-blocksize $\beta = 2 \cdot d$ may give a dimension too large for SVP to be tractable. However, by passing to a subfield, one can convert the instance to a high-rank problem at the cost of losing some algebraic structure – future research will build upon the analysis presented here to investigate this tradeoff.

2 Preliminaries

We write $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$ for the naturals, integers, rationals, reals, and complex numbers respectively, and $\mathbb{Z}_q$ for the integers modulo q. We write ln for the natural logarithm, ϕ for Euler's totient function, Γ for the Gamma function, and ψ for the digamma function. In Euclidean space $\mathbb{R}^n$ we write $B_n(r)$ for the n-ball of radius r, and $S_{n-1}(r)$ for the corresponding $(n-1)$-sphere. If r is omitted, it is understood to be 1.

2.1 Probability

We use $\mathbb{P}$, $\mathbb{V}$, and $\mathbb{E}$ to denote probability, variance, and expectation. For distributions $\mathcal{X}$, $\mathcal{Y}$ defined on a discrete set S, the statistical distance is $\Delta(\mathcal{X}, \mathcal{Y}) = \frac{1}{2}\sum_{s\in S}|\mathcal{X}(s) - \mathcal{Y}(s)|$. We use $\mathcal{X}^n$ to denote the distribution on S^n with components identically independently distributed (iid) according to $\mathcal{X}$. We write $\mathcal{N}(\mu, \sigma^2)$ for the normal (continuous Gaussian) distribution with mean μ and variance σ^2, $\mathrm{Bin}(n, p)$ for the binomial distribution with n trials and success probability p, $\mathcal{U}(S)$ for the uniform distribution on a finite nonempty set S, and $\mathrm{Beta}(a, b)$ for the beta distribution with shape parameters a and b. The sum of the squares of n independent $\mathcal{N}(0, \sigma^2)$ variables is the scaled chi-squared distribution $\sigma^2\chi^2_n$.

The discrete Gaussian on $\mathbb{Z}$ with mean 0 and parameter σ is defined by the probability mass function

$$D_{\mathbb{Z},\sigma}(x) = \frac{e^{-x^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-y^2/2\sigma^2}} \ .$$

Note that σ^2 does not correspond exactly to the variance of the distribution. However, for sufficiently large σ, many properties of the discrete Gaussian (including the variance) approach those of the continuous Gaussian, a phenomenon known as *smoothing* [MR07].

Definition 1 (Random Subspace). *A (uniformly) random β-dimensional subspace of $\mathbb{R}^N$ is sampled by choosing an arbitrary β-dimensional subspace $S \subset \mathbb{R}^N$ and applying a uniformly random orthogonal transformation $T \in O(N)$.*[3]

2.2 Lattices

Definition 2 (Lattice). *A rank n lattice is the integer span of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^d$, called a basis. We write these as columns of a matrix $\mathbf{B}$. The lattice generated by $\mathbf{B}$ is $\Lambda(\mathbf{B}) = \{x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n \mid x_i \in \mathbb{Z}\}$.*

Definition 3 (Gram–Schmidt Orthogonalisation, GSO). *For any lattice basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$, the Gram–Schmidt orthogonalisation is the set of orthogonal vectors denoted $\mathbf{b}^*_1, \ldots, \mathbf{b}^*_n$ and defined by*

$$\mathbf{b}^*_i = \mathbf{b}_i - \sum_{j<i} \mu_{i,j} \cdot \mathbf{b}^*_j \text{ where } \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}^*_j\rangle}{\langle \mathbf{b}^*_j, \mathbf{b}^*_j\rangle}.$$

Definition 4 (Lattice Volume). *Given any basis $\mathbf{B}$ for a lattice Λ, its volume is defined as the volume of the parallelepiped $\mathcal{P}(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in [0, 1)^n\}$,*

$$\mathrm{vol}(\Lambda) = \sqrt{\det\left(\mathbf{B}^T\mathbf{B}\right)} = \prod_{i=1}^{n} \|\mathbf{b}^*_i\|.$$

Definition 5 (Lattice Minima). *For a rank n lattice Λ we define, for each $i \in \{1, \ldots, n\}$, the ith minimum*

$$\lambda_i(\Lambda) = \arg\min_{r\in\mathbb{R}^+}\{\mathrm{rank}(\mathrm{span}_{\mathbb{R}}(\Lambda \cap B_n(r))) = i\}.$$

Of particular interest is the *first minimum* $\lambda_1(\Lambda)$, which is the length of a shortest (nonzero) vector in the lattice.

[3] Formally, the set of all β-dimensional subspaces of $\mathbb{R}^N$ is called the *Grassmannian manifold* and denoted $\mathrm{Gr}_\beta(\mathbb{R}^N)$. The orthogonal group $O(N)$, which acts transitively on $\mathrm{Gr}_\beta(\mathbb{R}^N)$, is compact and thus admits a unique finite *Haar measure*, which is a natural uniform measure on the group. The distribution we have defined on $\mathrm{Gr}_\beta(\mathbb{R}^N)$ is simply the *push-forward* of this Haar measure under the action of $O(N)$.

Definition 6 (Gaussian Heuristic). *For a rank n lattice of unit volume, the Gaussian heuristic for the shortest vector approximates the first minimum as*

$$\mathrm{gh}_{\mathbb{Q}}(n) = \frac{2^{\frac{1}{n}}\Gamma\left(1+\frac{1}{n}\right)}{\mathrm{vol}(B_n)^{\frac{1}{n}}} \approx \sqrt{\frac{n}{2\pi e}} \ .$$

By appropriate scaling, we define the Gaussian heuristic for a lattice of arbitrary volume, $\mathrm{gh}_{\mathbb{Q}}(\Lambda) = \mathrm{gh}_{\mathbb{Q}}(n) \cdot \mathrm{vol}(\Lambda)^{1/n}$. We write $\mathrm{lgh}_{\mathbb{Q}}(\cdot) = \ln \mathrm{gh}_{\mathbb{Q}}(\cdot)$, and may omit the subscript.

Definition 7 (Projected Sublattices). *Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis for a lattice Λ. We define the linear map π_i as the projection orthogonal to the span of $\{\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}\}$. For indices $1 \leq i < j \leq n$ we define the projected subbasis*

$$\mathbf{B}_{[i:j]} = (\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_j))$$

and the corresponding projected sublattice $\Lambda_{[i:j]} = \Lambda(\mathbf{B}_{[i:j]})$.

If the index j is omitted, we understand it to mean $j = n$. We may also write $\Lambda^{(1)}_{[i:j]}$ to denote the normalised projected sublattice, scaled to have volume 1.

2.3 Lattice Problems

Lattices are a rich source of computational problems. A simple problem is to find a shortest (nonzero) vector in a lattice; we further consider promise problem variants, where we are guaranteed that the lattice satisfies some additional property.

Definition 8 (Shortest Vector Problem, SVP). *Given a lattice Λ, find a vector $\mathbf{v} \in \Lambda$ of length $\lambda_1(\Lambda)$.*

Definition 9 (Unique Shortest Vector Problem, uSVP$_\gamma$). *Given a lattice Λ and $\gamma \geq 1$ with the promise that $\lambda_2(\Lambda) > \gamma\lambda_1(\Lambda)$, find the vector $\mathbf{v} \in \Lambda$ (unique up to sign) of length $\lambda_1(\Lambda)$. If omitted, $\gamma = 1$.*

While previously it was thought that the factor $\gamma = \lambda_2(\Lambda)/\lambda_1(\Lambda)$ determines the difficulty of solving SVP on a lattice, it is now understood that the hardness is mainly driven by the ratio $f = \mathrm{gh}(\Lambda)/\lambda_1(\Lambda)$ [AGVW17, BMW19, AD21].

Definition 10 (f-unusual-SVP [DvW22]). *Given a lattice Λ and $f \geq 1$ with the promise that $\lambda_1(\Lambda) \leq \mathrm{gh}(\Lambda)/f$, find a vector $\mathbf{v} \in \Lambda$ of length $\lambda_1(\Lambda)$.*

A commonly used problem in cryptography is *Learning with Errors* or LWE [Reg09], the presumed hardness of which is the underlying security assumption in many lattice-based schemes.

Definition 11 (LWE$_{n,m,q,\chi}$). *Let n, m, q be positive integers and χ a probability distribution on $\mathbb{Z}_q$. We call m the number of samples and q the modulus. Sample $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$, a secret vector $\mathbf{s} \leftarrow \chi^n$, an error vector $\mathbf{e} \leftarrow \chi^m$, and compute*

$$\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \pmod q$$

An instance of LWE$_{n,m,q,\chi}$ *is the pair $(\mathbf{A}, \mathbf{b})$ and the goal is to recover the secret vector $\mathbf{s}$.*

For cryptographic functionalities, the distribution χ should satisfy some notion of 'smallness', when considered as a distribution on $\mathbb{Z}$ using the balanced coset representatives $\{-\lceil q/2\rceil + 1, \ldots, \lfloor q/2\rfloor\}$. Often notation is abused to write χ as a probability distribution on $\mathbb{Z}$, with the understanding that we mean its reduction modulo q. Note that the given formulation, where $\mathbf{s}$ is sampled coordinatewise from χ, is known as *normal form* LWE. The original formulation samples $\mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, but a standard reduction [ACPS09] maps LWE with uniform $\mathbf{s}$ and m samples to normal form LWE with $m - n$ samples.

2.4 Lattice Reduction

A lattice reduction algorithm takes as input a basis for a lattice, and outputs a basis of better quality, called a reduced basis. The LLL algorithm [LLL82] achieves the following in polynomial time in its input.

Definition 12 (LLL-reduced). *For $\delta \in (1/4, 1)$ a basis $\mathbf{B}$ is δ-LLL reduced if $|\mu_{i,j}| \leq 1/2$ for all $1 \leq j < i \leq n$ and $(\delta - \mu_{i,i-1}^2)\|\mathbf{b}_{i-1}^*\|^2 \leq \|\mathbf{b}_i^*\|^2$ for all $i \in \{2, \ldots, n\}$. Unless otherwise specified, we assume $\delta = 0.99$.*

A stronger notion of reduction is achieved by the BKZ algorithm [Sch87, SE94].

Definition 13 (BKZ-reduced). *Let $2 \leq \beta \leq n$. A basis $\mathbf{B}$ of a rank n lattice Λ is* BKZ-β *reduced if it is LLL-reduced and for all $i \in \{1, \ldots, n-1\}$,*

$$\|\mathbf{b}_i^*\| = \lambda_1(\Lambda_{[i:\min(i+\beta-1,n)]})$$

The quantity β is called the blocksize. We say a basis is *HKZ-reduced* if it is BKZ-reduced with $\beta = n$. The BKZ algorithm requires access to an oracle O_{SVP} which solves SVP in lattices of rank at most β. This oracle is repeatedly called on projected sublattices $\Lambda_{[i:\min(i+\beta-1,n)]}$, known as *blocks*, and if the output vector $\mathbf{v}$ is shorter than the current first vector in the block, it is inserted into the basis at the beginning of the block. A high-level description is given in Algorithm 1.

Input : LLL reduced basis $\mathbf{B}$ of a rank n lattice Λ, blocksize β

repeat
  for $i \leftarrow 1$ **to** n **do**
    $\ell \leftarrow \|\mathbf{b}_i^*\|$
    $j \leftarrow \min(i + \beta - 1, n)$
    $\mathbf{v} \leftarrow O_{\mathrm{SVP}}(\pi_i(\Lambda_{[i:j]}))$
    if $\|\mathbf{v}\| \leq \ell$ **then**
      $\mathbf{v}' \leftarrow x_i\mathbf{b}_i + \cdots + x_j\mathbf{b}_j$ where $\mathbf{v} = x_i\pi_i(\mathbf{b}_i) + \cdots + x_j\pi_i(\mathbf{b}_j)$
      extend $\mathbf{B}$ by inserting $\mathbf{v}'$ at index i
      run LLL on $\mathbf{B}$ to remove linear dependencies, drop $\mathbf{0}$ column
    end
  end
until no insertions were made on previous tour;
return B

Algorithm 1: Simplified view of the BKZ algorithm. Each execution of the repeat context is called a *tour*.

BKZ in Practice. In its original form as presented in Algorithm 1, BKZ terminates only once no insertions were made during the previous tour, guaranteeing a BKZ-β reduced basis. In practice we do not run the algorithm to termination, but instead provide a fixed maximum number of tours τ. We also consider Progressive-BKZ, where we run a fixed number of tours τ at each blocksize $\beta = 3, 4, \ldots, n$ sequentially, or until some termination condition is met.[4]

2.5 Predicting Lattice Reduction

Definition 14 (Basis Profile). *For a basis $\mathbf{B}$ of a rank n lattice Λ with GSO $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$, the basis profile is the sequence $(\ell_i)_{i=1}^n$ where $\ell_i = \ln\|\mathbf{b}_i^*\|$.*

[4] In the context of the primal attack, when the embedded vector has been found.

A common heuristic, based on observed behaviour, is that after lattice reduction the profile ℓ_i decreases linearly.

Definition 15 (Geometric Series Assumption, GSA [Sch03]). *For β sufficiently large, there is a constant $\alpha > 1$ (depending only on β) such that the profile after BKZ-β reduction of a random full-rank n lattice basis satisfies*

$$\mathbb{E}[\ell_i] = \mathbb{E}[\ell_1] - (i-1)\ln\alpha \ .$$

Equivalently, for all $1 \leq i \leq n$,

$$\mathbb{E}[\ell_i] = \frac{n+1-2i}{2}\ln\alpha + \frac{1}{n}\ln \mathrm{vol}(\Lambda)$$

where

$$\ln\alpha = \frac{2}{\beta-1}\mathbb{E}_{\mathbf{s}}[\ln\|\mathbf{s}\|]$$

and $\mathbf{s}$ is a shortest vector in one of the random normalised projected sublattices $\Lambda^{(1)}_{[j:j+\beta-1]}$ for $1 \leq j \leq n-\beta+1$.

In the analysis of BKZ algorithms, it is standard to assume that for sufficiently large β the normalised projected sublattices behave as random unit volume lattices, and we hence apply the Gaussian heuristic.

Definition 16. *For $\beta \geq 45$, define α_β by*

$$\ln\alpha_\beta = \frac{2}{\beta-1}\cdot \mathrm{lgh}(\beta)$$

BKZ Simulation. The GSA only predicts the profile once sufficiently many tours have been run, and furthermore does not capture the true shape of the profile [AD21]. For refined predictions we may use *BKZ simulators*, which accept as input a basis profile $(\ell_i)_{i=1}^n$, a blocksize β, and number of tours τ, and the output is a prediction for the resulting profile. In this work we consider the deterministic simulator[5] introduced in [CN11], which we write as BKZSim.

2.6 Primal Attack & Blocksize Estimation

The *primal attack* refers to a family of algorithms for solving LWE which transform LWE instances into uSVP instances, and apply lattice reduction algorithms.

Definition 17 (Bai–Galbraith Embedding [BG14]). *Given an instance of $\mathrm{LWE}_{n,m,q,\chi}$, define the Bai–Galbraith embedding[6] as the block matrix*

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_m & -\mathbf{A}^T & \mathbf{b} \\ & \mathbf{I}_n & \\ & & t \end{pmatrix}$$

where $t \in \mathbb{R}$ is the embedding coefficient*. This gives a rank $N = (n+m+1)$ lattice $\Lambda = \Lambda(\mathbf{B})$ containing the vector $\mathbf{v} = (\mathbf{e}, \mathbf{s}, t)$.*

This is a specialisation of a more general method of Kannan [Kan87]. The vector $\mathbf{v}$ is called the *embedded vector*. Since $\mathbf{e}$ and $\mathbf{s}$ are sampled from a short distribution χ, the embedded vector is short, and under appropriate conditions is the unique shortest vector [LM09]. The embedding coefficient is often chosen in practice as $t = 1$, though formally one should choose $t = \sigma$ [LM09]. Indeed, if we view Λ as an instance of f-unusual-SVP, then σ is the choice maximising the expected value of the ratio $f = \mathrm{gh}(\Lambda)/\lambda_1(\Lambda)$.

[5] While there also exist probabilistic simulators [BSW18], we do not consider them in this work.

[6] We note that variants exist for LWE instances where $\mathbf{e}$ and $\mathbf{s}$ are sampled from different distributions, or when these distributions are not centred at zero [AGVW17, PV21], but we will not consider those scenarios.

Solving the uSVP instance. After forming the lattice $\Lambda = \Lambda(\mathbf{B})$ the primal attack aims to recover $\mathbf{v}$ by applying the BKZ algorithm. The mechanism by which BKZ recovers the short vector was suggested in [ADPS16] and studied further in [AGVW17, BMW19]. In brief, when the O_{SVP} oracle is called at position $i = N - \beta + 1$ (on the 'terminal block') the projection $\pi_{N-\beta+1}(\mathbf{v})$ of the embedded vector will be inserted if it is a shortest vector in the projected sublattice, leading to the success condition:

$$\|\pi_{N-\beta+1}(\mathbf{v})\| \leq \|\mathbf{b}^*_{N-\beta+1}\|. \tag{1}$$

If this occurs then, by a similar argument, in the following tour a projection of $\mathbf{v}$ will likely be inserted at position $i = N - 2\beta + 2$. Subsequent tours will continue to insert projections of $\mathbf{v}$ at earlier positions in the basis, until eventually the applications of LLL within BKZ are sufficient to recover $\mathbf{v}$ [AGVW17, §4.1].

Prediction via the GSA. An estimate for the smallest blocksize for which BKZ will recover the embedded vector, assuming the GSA, was given in [ADPS16] and is sometimes referred to as the '2016 estimate'. If $\sigma^2 = \mathbb{V}(\chi)$, then $\mathbb{E}[\|\mathbf{v}\|^2] = N\sigma^2$, and the projection into the heuristically random β-dimensional subspace spanned by $\Lambda_{[N-\beta+1]}$ has expected square norm $\mathbb{E}[\|\pi_{N-\beta+1}(\mathbf{v})\|^2] = \beta\sigma^2$.

On the other hand, after several tours of BKZ-β the profile converges towards the GSA, with the slope predicted by α_β. Thus, Equation (1) becomes

$$\sigma\sqrt{\beta} \leq \alpha_\beta^{(2\beta-N-1)/2}\mathrm{vol}(\Lambda)^{1/N} \tag{2}$$

and the first successful blocksize is estimated as the smallest β satisfying this inequality. If the attacker is free to choose the number of LWE samples m, then β and m may be optimised simultaneously.

Prediction via Simulation. The 2016 estimate has several shortcomings. Firstly, it assumes the GSA, which does not capture the true shape of the profile, and only applies when sufficiently many tours are run. Secondly, it only provides a single blocksize which is likely to be successful, but smaller blocksizes may still solve the instance.

A more fine-grained estimation technique, introduced in [DDGR20], models an attacker running Progressive-BKZ and estimates the probability mass function for the blocksize β^* at which the shortest vector will be recovered. The analogous model predicting the success probability of (fixed blocksize) BKZ-β, when running a maximum number of tours τ, was introduced in [PV21].

Input : $(n, m, q, \chi), \beta, \tau$

$p_{\mathrm{tot}} \leftarrow 0$, $\sigma^2 \leftarrow \mathbb{V}(\chi)$
$N \leftarrow n + m + 1$
$(\ell_i)_{i=1}^N \leftarrow \mathsf{LLLSim}(n, m, q, \chi)$
for tour $\leftarrow 1$ **to** τ **do**
  $(\ell_i)_{i=1}^N \leftarrow \mathsf{BKZSim}((\ell_i)_{i=1}^N, \beta, 1)$
  $p_{\mathrm{new}} \leftarrow \mathbb{P}\left(\sigma^2\chi_\beta^2 \leq \exp(2 \cdot \ell_{N-\beta+1})\right)$
  $p_{\mathrm{tot}} \leftarrow p_{\mathrm{tot}} + (1 - p_{\mathrm{tot}}) \cdot p_{\mathrm{new}}$
end
return p_{tot}

Algorithm 2: Success probability estimator for the primal attack running τ tours of BKZ-β, against $\mathrm{LWE}_{n,m,q,\chi}$, adapted from [PV21, Algorithm 6].

Input : $(n, m, q, \chi), \tau$

$p_{\text{tot}} \leftarrow 0$, $\sigma \leftarrow \mathbb{V}(\chi)$, $P \leftarrow \{\}$
$N \leftarrow n + m + 1$, $\beta \leftarrow 3$
$(\ell_i)_{i=1}^N \leftarrow \mathsf{LLLSim}(n, m, q, \chi)$
while $\beta < 40$ **do**
  $(\ell_i)_{i=1}^N \leftarrow \mathsf{BKZSim}((\ell_i)_{i=1}^N, \beta, \tau)$
  $\beta \leftarrow \beta + 1$
end
while $\beta \leq N$ **do**
  for `tour` $\leftarrow 1$ **to** τ **do**
    $(\ell_i)_{i=1}^N \leftarrow \mathsf{BKZSim}((\ell_i)_{i=1}^N, \beta, 1)$
    $p_{\text{new}} \leftarrow \mathbb{P}\left(\sigma^2 \chi_\beta^2 \leq \exp(2 \cdot \ell_{N-\beta+1})\right)$
    $p_{\text{tot}} \leftarrow p_{\text{tot}} + (1 - p_{\text{tot}}) \cdot p_{\text{new}}$
  end
  $P[\beta] \leftarrow p_{\text{tot}}$
  if $p_{\text{tot}} \geq 0.999$ **then**
    break
  end
  $\beta \leftarrow \beta + 1$
end
return P

Algorithm 3: Success probability estimator for the primal attack running Progressive-BKZ, with τ tours at each blocksize, against $\text{LWE}_{n,m,q,\chi}$, adapted from [PV21, Algorithm 5]. Returns the cumulative mass function $P[\beta]$ of solving the instance while using block size β.

The idea is to simulate the tour-by-tour evolution of the basis profile $(\ell_i)_{i=1}^N$ using BKZSim. After each tour it is assumed that the projection of the embedded vector into the terminal block has square norm following a chi-squared distribution,

$$\|\pi_{N-\beta+1}(\mathbf{v})\|^2 \sim \sigma^2 \chi_\beta^2 \tag{3}$$

and is inserted into the basis if shorter than $\mathbf{b}_{N-\beta+1}^*$, which occurs with probability $\mathbb{P}(\sigma^2\chi_\beta^2 \leq \exp(2 \cdot \ell_{N-\beta+1}))$. Note that this approach requires a prediction of the initial profile, when the Bai–Galbraith embedding $\mathbf{B}$ is only LLL-reduced. This exhibits a characteristic *Z-shape* [How07], which is well-understood. We write $\mathsf{LLLSim}(n, m, q, \chi)$ for the algorithm described in [PV21, App. B] which outputs the predicted Z-shape resulting from an instance of $\text{LWE}_{n,m,q,\chi}$.

For reference, we provide descriptions of estimators for the primal attack using BKZ (Algorithm 2) and Progressive-BKZ (Algorithm 3)

2.7 Number Fields

Let $K \cong \mathbb{Q}[X]/P(X)$ be a number field of degree $d = [K : \mathbb{Q}] = \deg P$. We write $\mathcal{O}_K$ for the ring of integers, Δ_K for the discriminant, and μ_K for the set of roots of unity. K admits d distinct embeddings (injective field homomorphisms) into $\mathbb{C}$, each one corresponding to evaluating polynomials in K at one of the roots of P in $\mathbb{C}$. We define the *canonical embedding* $\boldsymbol{\sigma} : K \to \mathbb{C}^d, x \mapsto (\sigma_i(x))_{i=1}^d$. We have $d = d_\mathbb{R} + 2d_\mathbb{C}$, where $d_\mathbb{R}$ denotes the number of real embeddings, and $d_\mathbb{C}$ the number of complex embeddings up to conjugation. We define the ring $K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R} \cong \mathbb{R}^{d_\mathbb{R}} \times \mathbb{C}^{d_\mathbb{C}}$, and for any $z \in K_\mathbb{R}$ define the complex conjugate $\bar{z} = \boldsymbol{\sigma}^{-1}(\overline{\boldsymbol{\sigma}(z)})$, and square root $z^{1/2} = \boldsymbol{\sigma}^{-1}(\boldsymbol{\sigma}(z)^{1/2})$, where $\boldsymbol{\sigma}$ has been extended to $K_\mathbb{R}$ and the conjugation and square root are applied componentwise.

The field trace $\mathrm{Tr} : K_{\mathbb{R}} \to \mathbb{R}, x \mapsto \sum_{j=1}^{d} \sigma_j(x)$ induces an inner product $\langle x, y\rangle_{\mathrm{Tr}} = \mathrm{Tr}(x\bar{y})$, which corresponds to $\langle \boldsymbol{\sigma}(x), \boldsymbol{\sigma}(y)\rangle_{\mathbb{C}}$. This extends to $K_{\mathbb{R}}^m$ by $\langle \mathbf{x}, \mathbf{y}\rangle_{\mathrm{Tr}} = \sum_{i=1}^{m} \langle x_i, y_i\rangle_{\mathrm{Tr}}$. On the other hand, we also define $\langle \mathbf{x}, \mathbf{y}\rangle_K = \sum_{i=1}^{m} x_i \bar{y}_i$. We write $\|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{x}\rangle_{\mathrm{Tr}}^{1/2}$ and $\|\mathbf{x}\|_K = \langle \mathbf{x}, \mathbf{x}\rangle_K^{1/2}$. The algebraic norm is $\mathrm{N}(x) = \prod_{i=1}^{d} \sigma_i(x)$ and satisfies, by the arithmetic-geometric inequality, $\sqrt{d}\mathrm{N}(x)^{1/d} \leq \|x\|$. For $\mathbf{x} \in K_{\mathbb{R}}^m$ we use the shorthand $\mathrm{N}(\mathbf{x}) = \mathrm{N}(\langle \mathbf{x}, \mathbf{x}\rangle_K^{1/2})$.

Cyclotomic Fields. Let $c \in \mathbb{N}$ be odd or a multiple of 4. The cyclotomic number field with *conductor* c is $K = \mathbb{Q}(\omega_c) \cong \mathbb{Q}[X]/\Phi_c(X)$, where ω_c is a primitive c-th root of unity, and Φ_c is the c-th cyclotomic polynomial. The degree is $[K : \mathbb{Q}] = \deg \Phi_c = \phi(c)$. We have $|\mu_K| = c$ when c is even, and $|\mu_K| = 2c$ when c is odd.

The Cyclic Embedding. For $K = \mathbb{Q}(\omega_c)$ a cyclotomic field, consider the *cyclic ring* $\mathcal{C} = \mathbb{R}[X]/(X^c - 1)$. We view $\mathcal{C}$ as a Euclidean vector space equipped with the *coefficient inner product* $\langle \sum_{i=0}^{c-1} a_i X^i, \sum_{i=0}^{c-1} b_i X^i\rangle_{\mathrm{coeff}} := \sum_{i=0}^{c-1} a_i b_i$. The appropriately scaled discrete Fourier transform

$$\mathcal{F} : \mathcal{C} \to \mathbb{C}^c, \quad f(X) \mapsto \frac{1}{\sqrt{c}} \left(f(\omega_c^j)\right)_{j=0}^{c-1}$$

provides an isometry with $\mathbb{C}^c$ (equipped with the standard inner product).

On the other hand, the canonical embedding $\boldsymbol{\sigma} : K \to \mathbb{C}^{\phi(c)}$ is an isometry (using the trace inner product on K), and there is a trivial isometry $\mathsf{pad} : \mathbb{C}^{\phi(c)} \to \mathbb{C}^c$ by null padding at positions corresponding to non-primitive c-th roots of unity.

Definition 18 (Cyclic Embedding). *In the notation above, the cyclic embedding $\theta : K \to \mathcal{C}$ is defined as the isometry completing the commutative diagram*

$$\begin{array}{ccc} K\,,\ \langle\cdot,\cdot\rangle_{\mathrm{Tr}} & \xrightarrow{\quad\boldsymbol{\sigma}\quad} & \mathbb{C}^{\phi(c)}\,,\ \langle\cdot,\cdot\rangle \\ \Big\downarrow{\scriptstyle \theta} & & \Big\downarrow{\scriptstyle \mathrm{pad}} \\ \mathcal{C}\,,\ \langle\cdot,\cdot\rangle_{\mathrm{coeff}} & \xrightarrow[\quad\mathcal{F}\quad]{} & \mathbb{C}^{c}\,,\ \langle\cdot,\cdot\rangle \end{array}$$

The cyclic embedding has been used, implicitly or explicitly, in several existing works [DD12, DP16, BDF18, DEP25]. The utility of the cyclic embedding in this paper is twofold. Firstly, the ring of integers $\mathcal{O}_K = \mathbb{Z}[\omega_c]$ is mapped under θ to elements of $\frac{1}{\sqrt{c}}\mathbb{Z}[X]/(X^c - 1)$;[7] hence, multiplying elements of K by a factor $\sqrt{c}$ results in a scaled isometric embedding of $\mathcal{O}_K$ in $\mathbb{Z}^c$. This representation allows us, in implementations, to view module lattices as integer lattices using the standard inner product – whereas using the canonical embedding would lead to irrational values, and using the coefficient embedding would require a different inner product.

Secondly, the cyclic embedding allows us to efficiently sample from distributions on $\mathcal{O}_K$ which are close to spherical with respect to the geometry defined by the trace inner product. This technique was introduced in [DD12], which concerns the primal formulation of Ring-LWE, and we will utilise it when generating instances of Module-LWE for experiments.

In particular, viewing $K = \mathbb{Q}[X]/\Phi_c(X)$, there is a canonical projection $\pi : \mathcal{C} \to K$ by reducing modulo Φ_c (note that for any x in K, $\theta(x) \bmod \Phi_c = x$). This induces a $\mathbb{R}$-linear map $\pi_{\mathbb{R}} : \mathcal{C} \to K_{\mathbb{R}}$. Furthermore, we define the rounding map $\lfloor\cdot\rceil : K_{\mathbb{R}} \to \mathcal{O}_K$ by rounding each coefficient, $\sum_{i=0}^{d-1} a_i X^i \to \sum_{i=0}^{d-1} \lfloor a_i \rceil X^i$.

[7] Let $a \in \mathcal{O}_K$. By definition, denoting $\theta(a) = \sum_{k=0}^{c-1} b_k X^k$, $b_k = \frac{1}{c}\sum_{i=1}^{d} \sqrt{c} \cdot \sigma_i(a) \cdot \sigma_i(\omega_c^{-k}) = \frac{1}{\sqrt{c}}\sum_{i=1}^{d} \sigma_i(a\omega_c^{-k}) = \frac{1}{\sqrt{c}}\mathrm{Tr}(a\omega_c^{-k})$. As both a and ω_c are in the ring $\mathcal{O}_K$, $a\omega_c^{-k}$ also is, and its trace is in $\mathbb{Z}$.

Definition 19. *With the notation above, we write χ_σ to mean the distribution on $\mathcal{O}_K$ sampled as follows.*

1. *Sample $g = \sum_{i=0}^{c-1} g_i X^i \in \mathcal{C}$ with each $g_i \leftarrow \mathcal{N}(0, \sigma^2/c)$*
2. *Return the rounded reduction $\lfloor \pi_{\mathbb{R}}(g) \rceil \in \mathcal{O}_K$.*

We note that $\pi_{\mathbb{R}}(g)$ is distributed as a continuous, spherical Gaussian with variance $d\sigma^2$ on $K_{\mathbb{R}}$ [DD12, Theorem 5]. However this is no longer true after the rounding step, with the distortion more pronounced for smaller σ. Nevertheless, for $a \leftarrow \chi_\sigma$ we have $\mathbb{E}[\|a\|^2] \approx d\sigma^2$, and similar to the integer discrete Gaussian $D_{\mathbb{Z},\sigma}$, this approximation improves for larger σ.

2.8 Module Lattices

A *fractional ideal* of $\mathcal{O}_K$ is an $\mathcal{O}_K$-submodule $\mathfrak{J} \subseteq K$ for which there exists $x \in K \setminus \{0\}$ such that $x\mathfrak{J} \subseteq \mathcal{O}_K$. The algebraic norm is $\mathrm{N}(\mathfrak{J}) = [\mathcal{O}_K : x\mathfrak{J}]/|\mathrm{N}(x)|$ for any such x. A rank r $\mathcal{O}_K$-module is a set of the form $\mathcal{M} = \sum_{i=1}^r \mathbf{b}_i \mathfrak{J}_i$ for nonzero fractional ideals $\mathfrak{J}_1, \ldots, \mathfrak{J}_r$ and $K_{\mathbb{R}}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_r \in K_{\mathbb{R}}^m$. $\mathcal{M}$ is said to be a *free* module if $\mathfrak{J}_1 = \cdots = \mathfrak{J}_r = \mathcal{O}_K$. The set $\mathfrak{B} = (\mathbf{b}_i, \mathfrak{J}_i)_{i=1}^r$ is a *pseudobasis* for $\mathcal{M}$, and called *unital* if $\mathcal{O}_K \subseteq \mathfrak{J}_i$ for all i. We write $\mathbf{B}$ for the matrix with columns $\mathbf{b}_1, \ldots, \mathbf{b}_r$. The module has $K_{\mathbb{R}}$-determinant $\det_{K_{\mathbb{R}}}(\mathcal{M}) = \det(\overline{\mathbf{B}}^T \mathbf{B}) \prod_{i=1}^r \mathfrak{J}_i$.

If we extend the canonical embedding $\boldsymbol{\sigma} : K \to \mathbb{C}^d$ componentwise to elements of $\mathcal{M}$, then $\boldsymbol{\sigma}(\mathcal{M})$ is an rd-dimensional lattice in $\mathbb{C}^{rd} \cong \mathbb{R}^{2rd}$, called a *module lattice*. We will abuse notation and write $\mathcal{M}$ to mean both the $\mathcal{O}_K$-module and the module lattice. The $\mathbb{Q}$-determinant is the volume of the module lattice, $\det_{\mathbb{Q}}(\mathcal{M}) = |\Delta_K|^{r/2} \mathrm{N}(\det_{K_{\mathbb{R}}}(\mathcal{M}))$.

It has been shown, under certain conditions on K and r, that the first minimum of a random (formally defined using the Haar measure) unit volume rank r module lattice over a number field K is asymptotically concentrated around $(|\mu_K| \cdot \mathrm{vol}(B_{rd}))^{-rd}$ [GSVV24]. For cyclotomic K, this was adapted to give the *module Gaussian heuristic*, and justified experimentally, in [DEP25].

Definition 20 (Module Gaussian Heuristic). *Let $\mathcal{M}$ be a rank r module lattice of unit volume over $K = \mathbb{Q}(\omega_c)$. The Gaussian heuristic for $\mathcal{M}$ approximates the first minimum as*

$$\mathrm{gh}_K(r) = \mathrm{gh}_{\mathbb{Q}}(rd) \cdot \left(\frac{|\mu_K|}{2}\right)^{1/rd}.$$

We write $\mathrm{lgh}_K(\cdot) = \ln \mathrm{gh}_K(\cdot)$. Again, by appropriate scaling, we define the module Gaussian heuristic for arbitrary $\mathcal{M}$ as $\mathrm{gh}_K(\mathcal{M}) = \mathrm{gh}_K(r) \cdot \det_{\mathbb{Q}}(\mathcal{M})^{1/rd}$.

Module-LWE provides another hardness assumption for cryptography [BGV12, LS15], and the algebraic structure can be utilised for better efficiency and smaller public key sizes, compared to unstructured LWE.

Definition 21 ($\mathrm{MLWE}_{K,r,m,q,\chi}$). *Let r, m, q be positive integers and χ a probability distribution on $R_q = \mathcal{O}_K/q\mathcal{O}_K$. Sample $\mathbf{A} \leftarrow \mathcal{U}(R_q^{r \times m})$, a secret vector $\mathbf{s} \leftarrow \chi^r$, and error vector $\mathbf{e} \leftarrow \chi^m$, and compute*

$$\mathbf{b} = \mathbf{A}^T \mathbf{s} + \mathbf{e} \pmod{q\mathcal{O}_K}$$

An instance of $\mathrm{MLWE}_{K,r,m,q,\chi}$ is the pair $(\mathbf{A}, \mathbf{b})$ and the goal is to recover the secret vector $\mathbf{s}$.

Again, we may abuse notation and give χ as a distribution on $\mathcal{O}_K$, with the understanding that we mean its reduction modulo $q\mathcal{O}_K$.

Note that formally the problem should be defined using the *dual ring* $R_q^\vee$ [LPR13], but we have chosen to use the *primal* formulation, as is commonly done for practical purposes [DD12, BJRW23]. As with LWE we consider *normal form* MLWE, noting that recent works [BJTW25] extend the standard reduction [ACPS09] to the module setting.

2.9 Module Lattice Reduction

Definition 22 (GSO over $K_{\mathbb{R}}$). *Given $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_r)$ which are $K_{\mathbb{R}}$-linearly independent, the GSO is*

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j<i} \mu_{i,j}\mathbf{b}_j^* \text{ where } \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle_K}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle_K}$$

As in the unstructured case, for a pseudobasis $\mathfrak{B} = (\mathbf{b}_i, \mathfrak{J}_i)_{i=1}^r$ of a module lattice $\mathcal{M}$ we define π_i as the $K_{\mathbb{R}}$-linear projection away from the span of $\{\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}\}$.

For $1 \le i < j \le r$ we define the *projected subbasis* $\mathfrak{B}_{[i:j]} = (\pi_i(\mathbf{b}_k), \mathfrak{J}_k)_{k=i}^j$ and the corresponding *projected submodule* $\mathcal{M}_{[i:j]} = \pi_i(\mathbf{b}_i)\mathfrak{J}_i + \ldots + \pi_i(\mathbf{b}_j)\mathfrak{J}_j$.

Definition 23 (K-profile). *The pseudobasis $\mathfrak{B} = (\mathbf{b}_i, \mathfrak{J}_i)_{i=1}^r$ has K-profile $(\ell_i^K)_{i=1}^r$ where $\ell_i^K = \ln \det_{\mathbb{Q}}(\mathbf{b}_i^* \cdot \mathfrak{J}_i)$.*

Definition 24 ($\mathrm{mBKZ}_K^{\beta_K}$-reduced). *Let $2 \le \beta_K \le r$. A pseudobasis $\mathfrak{B}$ of a rank r module $\mathcal{M}$ is $\mathrm{mBKZ}_K^{\beta_K}$-reduced if for all $i \in \{1, \ldots, r-1\}$,*

$$\|\mathbf{b}_i^*\| = \lambda_1(\mathcal{M}_{[i:\min(i+\beta_K-1,r)]}) \ .$$

To achieve this, we apply the Module-BKZ algorithm, an overview of which is given in Algorithm 4. In practice we will consider running $\mathrm{mBKZ}_K^{\beta_K}$ for a fixed number of tours τ, and also consider Progressive-mBKZ_K which runs τ tours at each blocksize $\beta_K = 3, 4, \ldots$ sequentially.

Input : Unital pseudobasis $\mathfrak{B}$ of a rank r module $\mathcal{M}$, blocksize β_K

repeat
  for $i \leftarrow 1$ **to** r **do**
    $j \leftarrow \min(i + \beta_K - 1, r)$
    $\mathbf{v} \leftarrow O_{\mathrm{SVP}}(\mathcal{M}_{[i:j]})$
    Let $\mathfrak{J}$ be such that $\mathbf{v}\mathfrak{J} = \mathbf{v}K \cap \mathcal{M}_{[i:j]}$
    Lift $\mathbf{v}\mathfrak{J}$ to $\mathbf{w}\mathfrak{J} \subseteq \mathcal{M}$ satisfying $\pi_i(\mathbf{w}) = \mathbf{v}$
    Insert $(\mathbf{w}, \mathfrak{J})$ into $\mathfrak{B}$ at position i and remove linear dependencies.
  end
until no insertions were made on previous tour;
return $\mathfrak{B}$

Algorithm 4: Simplified view of the Module-BKZ algorithm, denoted $\mathrm{mBKZ}_K^{\beta_K}$, adapted from [DEP25, Algorithm 1].

Note that mBKZ using the K-blocksize β_K calls the SVP oracle on lattices of $\mathbb{Q}$-dimension at most $\beta = d\beta_K$, which we call the $\mathbb{Q}$-blocksize.

Definition 25 (Module-GSA, mGSA [DEP25]). *Let K be a number field of degree d. Let $\mathcal{M}$ be a random rank r module over K, and let $\mathfrak{B}$ be an $\mathrm{mBKZ}_K^{\beta_K}$-reduced pseudobasis of $\mathcal{M}$ for some sufficiently large $\beta_K \ll r$. Then there is a constant $\alpha_K > 1$ (depending on β_K) such that $\mathbb{E}[\ell_i^K] = \mathbb{E}[\ell_1^K] - (i-1)\ln \alpha_K$.*

Equivalently, $\mathbb{E}[\ell_i^K] = \frac{r+1-2i}{2} \ln \alpha_K + \frac{1}{r} \ln \det_{\mathbb{Q}}(\mathcal{M})$ for all $1 \le i \le r$, with

$$\ln \alpha_K = \frac{2d}{\beta_K - 1} \left(\mathbb{E}_{\mathbf{s}}[\ln \|\mathbf{s}\|] + \frac{1}{2d} \ln \frac{|\Delta_K|}{d^d} + \mathbb{E}_{\mathbf{s}} \left[\ln \frac{\sqrt{d}\mathrm{N}(\mathbf{s})^{1/d}}{\|\mathbf{s}\|} \right] + \frac{\mathbb{E}_{\mathfrak{J}}[\ln \mathrm{N}(\mathfrak{J})]}{d} \right)$$

where $\mathbf{s}$ is a shortest vector in one of the random normalised projected module lattices $\mathcal{M}^{(1)}_{[j:j+\beta_K-1]}$ for $1 \le j \le r - \beta_K + 1$, and $\mathfrak{J}$ is the corresponding fractional ideal in the random

unital pseudobasis $\mathfrak{B}$.[8] We write $\text{slope}_K(\beta_K) = -\ln \alpha_K$. The four terms in the expression for $\ln \alpha_K$ are examined in [DEP25, §4], under the further heuristic assumption that the projected sublattices behave as random unit volume lattices. The first may be estimated by the module-lattice Gaussian heuristic as $\text{lgh}_K(\beta_K)$. The second depends only on K and is called the *discriminant gap*. The third is the *skewness gap* and modelled as $\text{skew}_K(\beta_K)$ where

$$\text{skew}_K(r) = \frac{\ln d}{2} + \frac{d_{\mathbb{R}}\psi(r/2) + 2d_{\mathbb{C}}(\psi(r) - \ln 2)}{2d} - \frac{\psi(rd)}{2}$$

Finally, the fourth term is the *index gap*. Under suitable heuristics this was shown to be exponentially small, and in practice is almost always zero for the lattices and blocksizes we consider [DEP25, §4.5]. Moreover, embeddings of MLWE lattices are free, hence falling in the principal ideal class, and our experiments take as input bases with all fractional ideals $\mathfrak{J}_1, \ldots, \mathfrak{J}_r$ equal to $\mathcal{O}_K$ instead of more general pseudobases.

3 The Success Probability of Solving SVP via BKZ

In this section, we propose a new model for the success probability of solving SVP via BKZ. We apply our model to LWE, comparing to extensive experimental data, and demonstrate that it is not only more accurate than the existing estimators in [PV21], but also capable of capturing the difference in security between different distributions χ.

3.1 The Beta Distribution Model

We have seen that existing models for estimating the success probability (Algorithms 2 and 3) apply the heuristic assumption that, after each tour, the projection of the embedded vector in the terminal block has square norm following a chi-squared distribution. Indeed, modelling $S = \text{span}_{\mathbb{R}}(\Lambda_{[N-\beta+1]}) \subset \mathbb{R}^N$ as a random[9] β-dimensional subspace, and modelling $\mathbf{v}$ as sampled coordinatewise from a continuous Gaussian with variance σ^2, it follows that $\|\pi_{N-\beta+1}(\mathbf{v})\|^2 \sim \sigma^2\chi^2_\beta$.

However, this reasoning implicitly assumes that $\mathbf{v}$ is rerandomised each tour. In reality, when attempting to solve a single instance, $\mathbf{v}$ is fixed – it is only the subspace S which is rerandomised each tour.

Lemma 1. *Let $\mathbf{v} \in \mathbb{R}^N$ be arbitrary, and let $S \subset \mathbb{R}^N$ be a uniformly random β-dimensional subspace. The projection of $\mathbf{v}$ onto S, denoted $\pi_S(\mathbf{v})$, has square norm following a beta distribution:*

$$\|\pi_S(\mathbf{v})\|^2 \sim \|\mathbf{v}\|^2 \cdot \text{Beta}\left(\frac{\beta}{2}, \frac{N-\beta}{2}\right).$$

Proof. See Appendix A.1.

As a consequence of Lemma 1 and the preceding discussion, we propose new estimators for the success probability of solving SVP via BKZ, essentially replacing the chi-squared distributions in Algorithms 2 and 3 with the appropriate beta distributions. We have also decided to phrase our algorithms for a general lattice Λ, rather than one arising via the primal attack applied to an LWE instance. We require as input two properties of the lattice: an initial basis profile, and the first minimum λ_1. While in general computing λ_1 is a hard problem, for LWE lattices λ_1 is a random variable following a known distribution, and we will later integrate over this distribution (§3.2). The new estimators BKZSuccess and ProgBKZSuccess are described in Algorithms 5 and 6.

[8] As $\mathbf{s}$ is a shortest vector in the module, $\mathcal{O}_K \subseteq \mathfrak{J}$. The rest of the pseudobasis is made unital by applying [LPSW19, Alg. 3.2] in the mBKZ reduction.

[9] As per Definition 1.

$\mathsf{BKZSuccess}((\ell_i)_{i=1}^N, \lambda_1, \beta, \tau)$

$p_{\mathrm{tot}} \leftarrow 0$
for `tour` $\leftarrow 1$ **to** τ **do**
  $(\ell_i)_{i=1}^N \leftarrow \mathsf{BKZSim}((\ell_i)_{i=1}^N, \beta, 1)$
  $p_{\mathrm{new}} \leftarrow \mathbb{P}\big(\lambda_1^2 \cdot \mathrm{Beta}\big(\frac{\beta}{2}, \frac{N-\beta}{2}\big) \leq \exp(2 \cdot \ell_{N-\beta+1})\big)$
  $p_{\mathrm{tot}} \leftarrow p_{\mathrm{tot}} + (1 - p_{\mathrm{tot}}) \cdot p_{\mathrm{new}}$
end
return p_{tot}

Algorithm 5: Proposed estimator for the success probability of solving SVP via τ tours of BKZ-β.

$\mathsf{ProgBKZSuccess}\big((\ell_i)_{i=1}^N, \lambda_1, \tau\big)$

$p_{\mathrm{tot}} \leftarrow 0$, $P \leftarrow \{\}$, $\beta \leftarrow 3$
while $\beta < 40$ **do**
  $(\ell_i)_{i=1}^N \leftarrow \mathsf{BKZSim}((\ell_i)_{i=1}^N, \beta, \tau)$
  $\beta \leftarrow \beta + 1$
end
while $\beta \leq N$ **do**
  for `tour` $\leftarrow 1$ **to** τ **do**
    $(\ell_i)_{i=1}^N \leftarrow \mathsf{BKZSim}((\ell_i)_{i=1}^N, \beta, 1)$
    $p_{\mathrm{new}} \leftarrow \mathbb{P}\big(\lambda_1^2 \cdot \mathrm{Beta}\big(\frac{\beta}{2}, \frac{N-\beta}{2}\big) \leq \exp(2 \cdot \ell_{N-\beta+1})\big)$
    $p_{\mathrm{tot}} \leftarrow p_{\mathrm{tot}} + (1 - p_{\mathrm{tot}}) \cdot p_{\mathrm{new}}$
  end
  $P[\beta] \leftarrow p_{\mathrm{tot}}$
  if $p_{tot} \geq 0.999$ **then**
    break
  end
  $\beta \leftarrow \beta + 1$
end
return P

Algorithm 6: Proposed estimator for the success probability of solving uSVP via Progressive-BKZ, with τ tours at each blocksize. Returns the cumulative mass function $P[\beta]$ of solving the instance in the round using block size β.

3.2 Application to LWE

To apply these estimators to the case of $\mathrm{LWE}_{n,m,q,\chi}$, where we reduce the Bai-Galbraith lattice Λ with embedded vector $\mathbf{v}$, we use for the initial profile the predicted Z-shape $(\ell_i)_{i=1}^N = \mathsf{LLLSim}(n, m, q, \chi)$. However, the first minimum $\lambda_1(\Lambda)$, which is the norm of the embedded vector $\mathbf{v}$, is not deterministic: it is a random variable depending on the LWE instance. We capture this as follows (choosing $\mathsf{BKZSuccess}$ for illustration). Assuming the probability density function f_V of the random variable $V = \|\mathbf{v}\|^2$ is known, we estimate the success probability against a random $\mathrm{LWE}_{n,m,q,\chi}$ instance by computing the integral

$$p = \int_{\mathrm{Supp}(V)} \mathsf{BKZSuccess}((\ell_i)_{i=1}^N, \sqrt{v}, \beta, \tau) \cdot f_V(v)\,\mathrm{d}v\,.$$

In practice, χ is typically a discrete probability distribution, thus $\mathrm{Supp}(V)$ is a discrete set, and the integral becomes a sum:

$$p = \sum_{v \in \mathrm{Supp}(V)} \mathsf{BKZSuccess}((\ell_i)_{i=1}^N, \sqrt{v}, \beta, \tau) \cdot \mathbb{P}(\|\mathbf{v}\|^2 = v)\,.$$

We compute p by computing each term numerically.[10] The first factor is given by the output of Algorithm 5.[11] Recalling that the embedded vector is given by $\mathbf{v} = (\mathbf{e}, \mathbf{s}, t)$, where $\mathbf{e} \leftarrow \chi^m$, $\mathbf{s} \leftarrow \chi^n$ and t is the embedding coefficient, we compute the second factor exactly for certain distributions χ, and approximate it for others.

- *Centred Binary:* If $\chi = \mathcal{U}(\{-1, 1\})$ then $\|\mathbf{v}\|^2 = n + m + t^2$ is constant, and so the sum contains only one term.
- *Ternary:* If $\chi = \mathcal{U}(\{-1, 0, 1\})$ then $\|\mathbf{v}\|^2 \sim t^2 + \mathrm{Bin}(n + m, 2/3)$.
- *Discrete Gaussian:* If $\chi = D_{\mathbb{Z},\sigma}$ then it is difficult to capture the distribution of $\|\mathbf{v}\|^2$ exactly. However, note that if χ were a *continuous* Gaussian $\chi = \mathcal{N}(0, \sigma^2)$ we would have $\|\mathbf{v}\|^2 \sim \sigma^2 \chi^2_{n+m} + t^2$. Thus we make the following approximationwhich we examine further in Appendix A.2,

$$\mathbb{P}(\|\mathbf{v}\|^2 = v) \approx \mathbb{P}\left(v - \frac{1}{2} \leq \sigma^2 \chi^2_{n+m} + t^2 < v + \frac{1}{2}\right).$$

Furthermore, we note that we need not restrict ourselves, as we have done so far, to the regime where the LWE secret and error are sampled coordinatewise from some distribution χ. The model is far more general and can easily be adapted to, for example, sparse binary or ternary secrets.

3.3 Comparison to Experiments

To assess the accuracy of our new estimators, we compare to extensive experimental data.

n	q	m	σ	β_{GSA}
93	257	105	1	61
100	257	104	$\sqrt{2/3}$	60

Table 1: LWE parameter sets used for testing the primal attack and success probability simulators, also used in [PV21].

LWE Parameters. We consider LWE with two parameter sets, listed in Table 1. Each parameter set consists of n, q, m and σ^2, as well as the blocksize prediction β_{GSA} given by the 2016 estimate. For both parameter sets, we consider the discrete Gaussian distribution $\chi = D_{\mathbb{Z},\sigma}$. For the $n = 93$ parameter set, where $\sigma = 1$, we may also consider the centred binary distribution $\chi = \mathcal{U}(\{-1, 1\})$. For the $n = 100$ parameter set, where $\sigma = \sqrt{2/3}$, we may consider for comparison the ternary distribution $\chi = \mathcal{U}(\{-1, 0, 1\})$.[12]

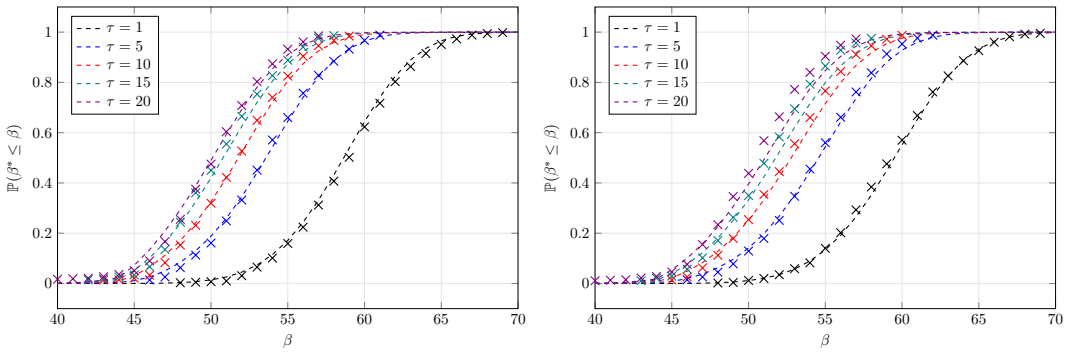
Implementation Details. We provide implementations in Python of Algorithms 5 and 6, and their applications to LWE as detailed in §3.2. We implement the primal attacks using BKZ 2.0 as provided in `fpylll` [dt25a].

We note that the runtime of our new estimators is not noticeably different to the original estimators Algorithms 2 and 3. Indeed, the runtime is dominated by the BKZ simulator, which remains unchanged.

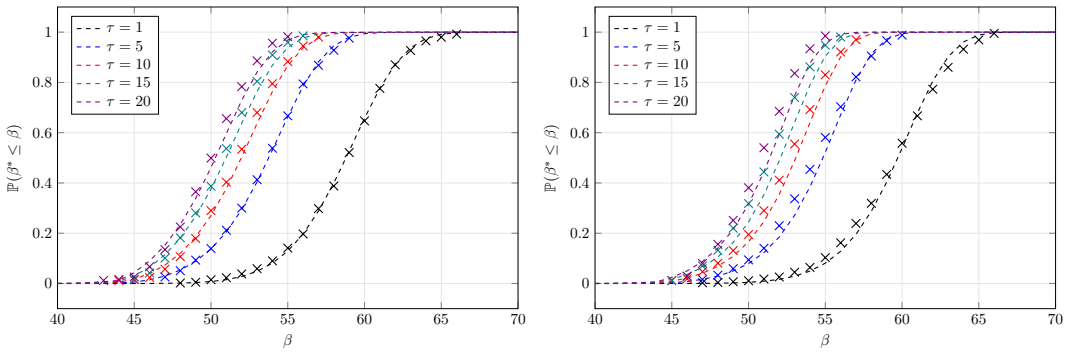
[10] If $\mathrm{Supp}(V)$ is infinite, we reduce to only finitely many terms by considering appropriate tail bounds on the distribution.

[11] We note that the calls to BKZSim within Algorithm 5 are independent of the input $\lambda_1 = v$, hence the profile simulation can be amortised.

[12] While the variance of $D_{\mathbb{Z},\sigma}$ may differ from σ^2, since we are not ensuring any conditions relating to smoothing, these examples were chosen in [PV21] to give different distributions in the same 'ballpark'.



(a) $n = 100, \chi = D_{\mathbb{Z},\sigma}$ **(b)** $n = 93, \chi = D_{\mathbb{Z},\sigma}$



(c) $n = 100, \chi = \mathcal{U}(\{-1, 0, 1\})$ **(d)** $n = 93, \chi = \mathcal{U}(\{-1, 1\})$

Figure 1: Comparison of estimated and experimental success probabilities for Progressive-BKZ. Dashed lines are estimations, crosses are experimental data. For each subfigure, 1000 experiments were run. See Table 1 for full parameter sets.

We also note that the success probabilities (both experimental and estimated) are higher than those reported in the published version of [PV21], due to a slightly different Bai–Galbraith embedding; it is now standard practice to place the q-vectors at the *start* of the basis, as opposed to the middle, which gives a more extreme Z-shape. Their estimators have been updated to account for this improvement, allowing for fair comparison to our estimators.

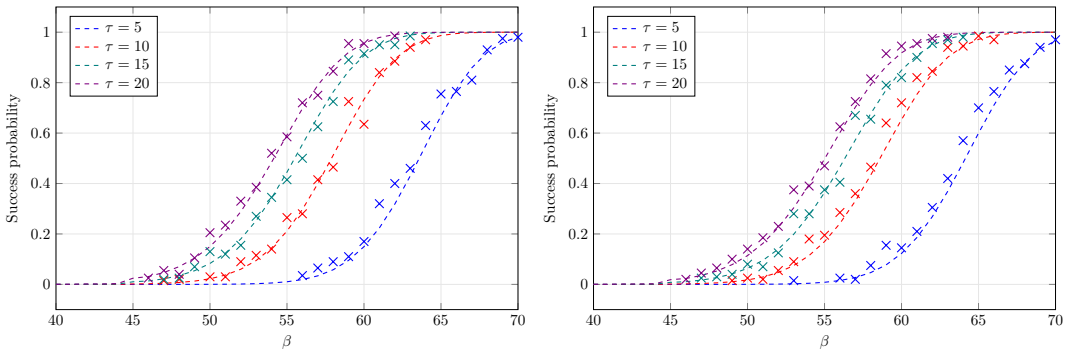
Measuring Accuracy. To evaluate our estimators against those from [PV21], we compute the total squared error between estimated and experimental probabilities.

For Progressive-BKZ, the estimators give a prediction for the cumulative mass function $\mathbb{P}(\beta^* \leq \beta)$, where β^* is the blocksize at which the short vector is recovered. From this we may extract a probability mass function $f_{\text{est}}(\beta)$. On the other hand, by counting how many experiments succeed at each blocksize, we obtain an experimental probability mass function $f_{\text{exp}}(\beta)$. The total square error is then $\sum_\beta (f_{\text{est}}(\beta) - f_{\text{exp}}(\beta))^2$.

For BKZ-β, we obtain for each β an estimate $f_{\text{est}}(\beta)$ of the success probability when using β, and an experimental probability $f_{\text{exp}}(\beta)$. The total square error is again $\sum_\beta (f_{\text{est}}(\beta) - f_{\text{exp}}(\beta))^2$.

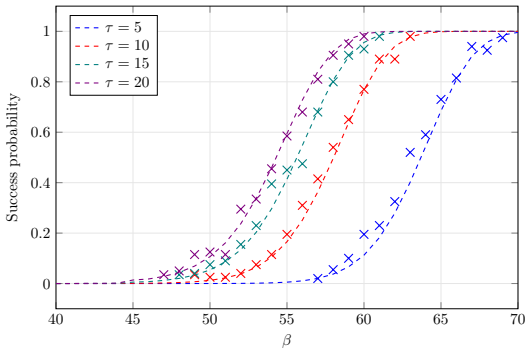
In each case we also provide a statistical baseline, to understand when the distances become significant. In particular, we compute the expected total square error if the same number of samples were drawn again from the experimentally obtained distribution and used to estimate it, as explained in Appendix B.1.

Furthermore, in Appendix B.2 we perform the same analysis instead using the statistical distance, an alternative measure of statistical closeness widely used in cryptology.

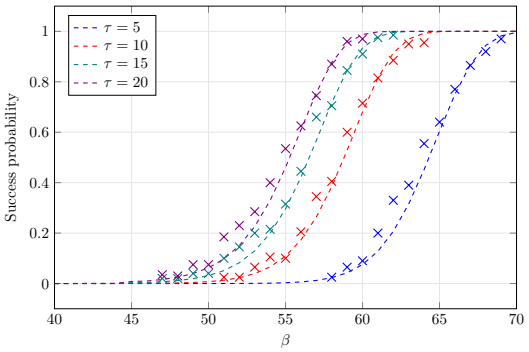


(a) $n = 100, \chi = D_{\mathbb{Z},\sigma}$

(b) $n = 93, \chi = D_{\mathbb{Z},\sigma}$



(c) $n = 100, \chi = \mathcal{U}(\{-1, 0, 1\})$



(d) $n = 93, \chi = \mathcal{U}(\{-1, 1\})$

Figure 2: Comparison of estimated and experimental success probabilities for BKZ-β. Dashed lines are estimations, crosses are experimental data. For each blocksize in each subfigure, 200 experiments were run. See Table 1 for full parameter sets.

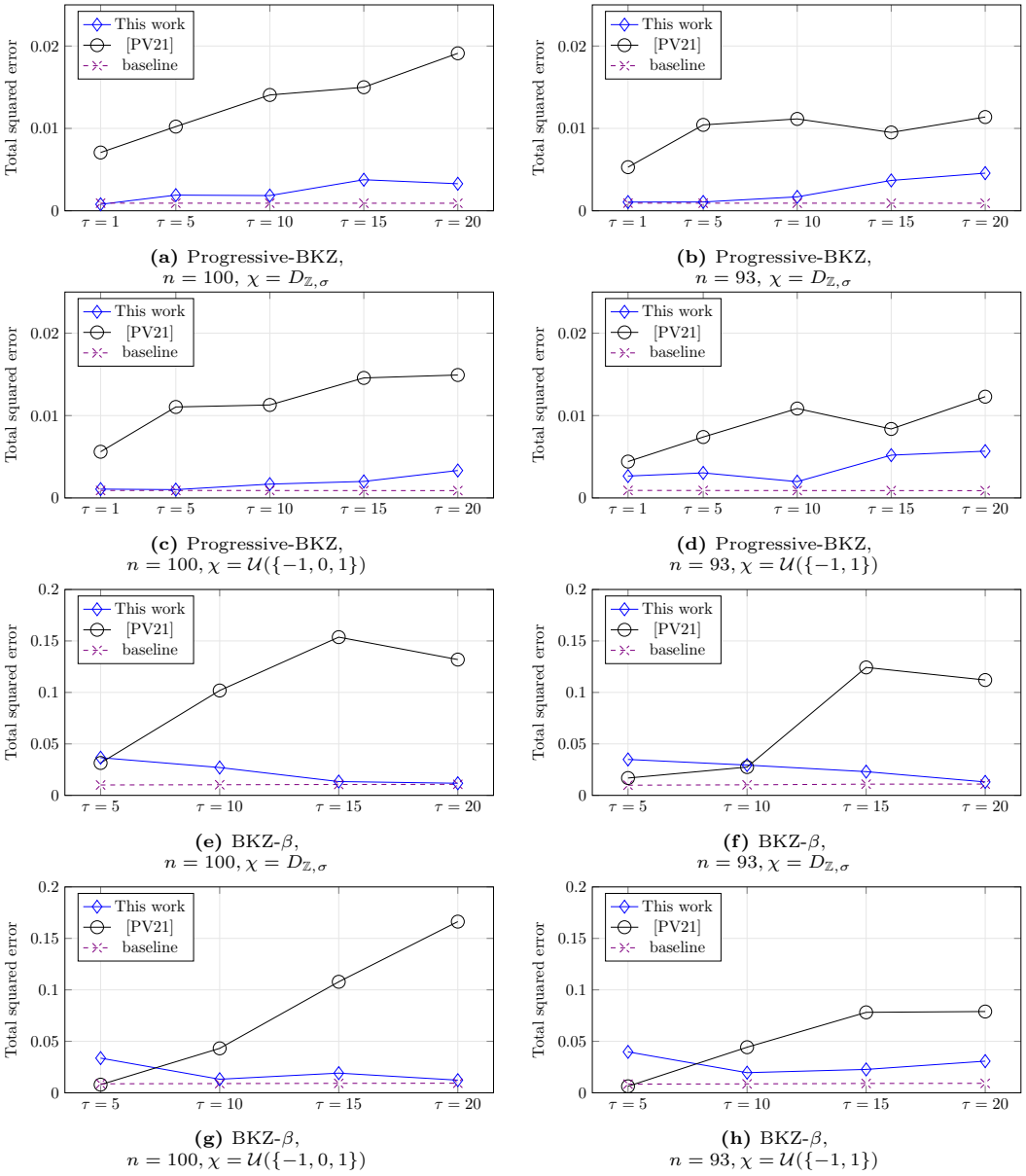
Results & Observations. In Figures 1 and 2 we show the output of our new estimators alongside experimental success probabilities. In Figure 3 we show, for both our new estimators and the old [PV21] estimators, the total squared error to the experiments. To illustrate the qualitative difference between the old and new estimators, we compare their outputs in Figure 4. Finally, in Figure 5, we illustrate the difference in success probability for the various error and secret dsitributions (centred binary, ternary, and discrete Gaussian).

From Figure 3 we see that our new model provides a consistent improvement in accuracy for Progressive-BKZ. In the case of BKZ-β, our model is marginally worse for $\tau = 5$ tours, but superior in all other cases.

In particular, when a larger number of tours are run, the [PV21] estimators vastly overestimate the success probability (see Figure 4). This is due to implicitly assuming the embedded vector is sampled freshly each tour, and hence it always has a chance to be unusually small. These probabilities accumulate each tour, making the effect more pronounced when more tours are run. Our new model eliminates this flaw by design.

Furthermore, our new model is able to faithfully capture the difference in security between the different secret and error distributions (Figure 5), whereas the old estimators are blind to the difference.

However, our new estimators often slightly *underestimate* the success probability. The cause of this is unclear; it is possibly due to the BKZ 2.0 algorithm in practice applying certain improvements to the basis quality (local block rerandomisation and preprocessing) which are not captured by the model.



(a) Progressive-BKZ, $n = 100$, $\chi = D_{\mathbb{Z},\sigma}$

(b) Progressive-BKZ, $n = 93$, $\chi = D_{\mathbb{Z},\sigma}$

(c) Progressive-BKZ, $n = 100, \chi = \mathcal{U}(\{-1, 0, 1\})$

(d) Progressive-BKZ, $n = 93, \chi = \mathcal{U}(\{-1, 1\})$

(e) BKZ-β, $n = 100, \chi = D_{\mathbb{Z},\sigma}$

(f) BKZ-β, $n = 93, \chi = D_{\mathbb{Z},\sigma}$

(g) BKZ-β, $n = 100, \chi = \mathcal{U}(\{-1, 0, 1\})$

(h) BKZ-β, $n = 93, \chi = \mathcal{U}(\{-1, 1\})$

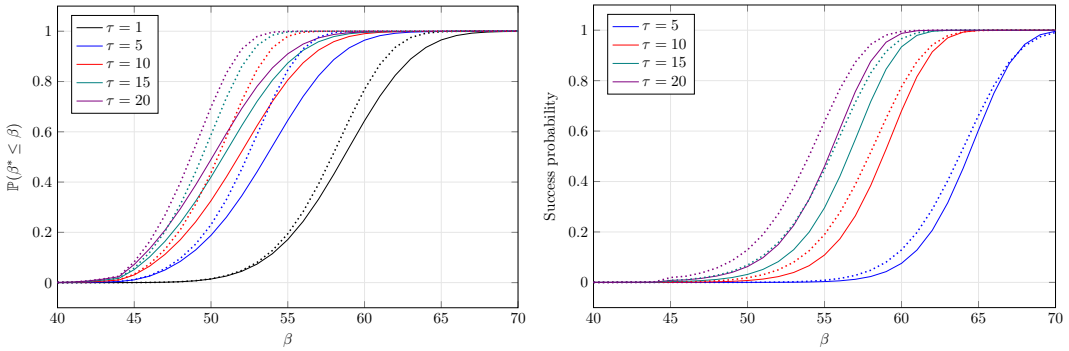
Figure 3: Total squared error between estimated and experimental success probabilities, compared to a baseline of the expected total squared error if drawing the same number of samples again from the experimental distribution (see Appendix B.1 for details).

4 Module-BKZ Profile Simulation

In this section, we propose an algorithm to simulate the evolution of the K-profile of a pseudobasis when running Module-BKZ, and we verify its accuracy experimentally.

4.1 Algorithm Description

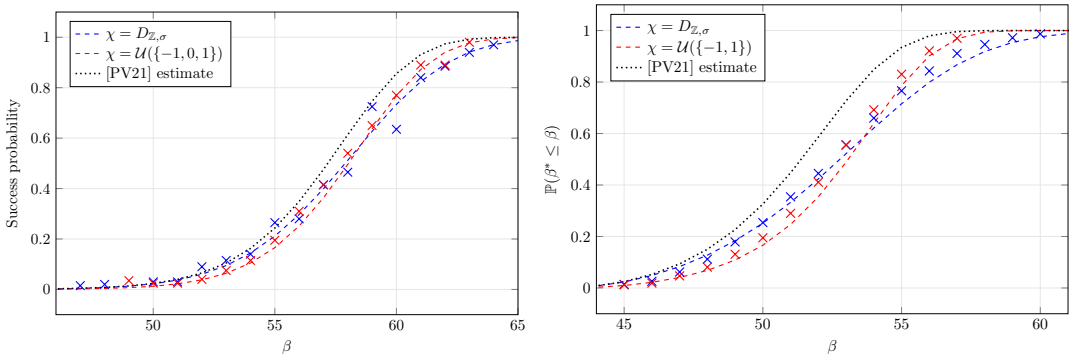
Suppose we have a pseudobasis $\mathfrak{B}$ with K-profile $(\ell_i)_{i=1}^{\rho}$. During a tour of $\text{mBKZ}_K^{\beta_K}$, when the SVP oracle is called at position i, it will find the shortest vector $\mathbf{w}$ in the projected module lattice $\mathcal{M}_{[i:i+b-1]}$ where $b = \min(\beta_K, \rho - i + 1)$ is the K-rank of the local block.



(a) Progressive-BKZ, $n = 100, \chi = D_{\mathbb{Z},\sigma}$

(b) BKZ-β, $n = 93, \chi = \mathcal{U}(\{-1, 1\})$

Figure 4: Comparison between our estimators (solid) and those in [PV21] (dotted).



(a) BKZ-β, $\tau = 10$, $n = 100$

(b) Progressive-BKZ, $\tau = 10$, $n = 93$

Figure 5: Comparison between the various secret and error distributions. Crosses are experiments, dashes are our estimators, dots are [PV21] estimators.

Then, the corresponding ideal $\mathfrak{J}$ is computed, and $(\mathbf{w}, \mathfrak{J})$ is inserted into the pseudobasis $\mathfrak{B}$ at position i.

Following this insertion operation, the updated value of the K-profile at position i, which we denote by ℓ'_i, is thus given by:

$$\begin{aligned}\ell'_i &= \ln \det_{\mathbb{Q}}(\mathbf{w}\mathfrak{J}) = \frac{1}{2}\ln|\Delta_K| + \ln \mathrm{N}(\mathbf{w}) + \ln \mathrm{N}(\mathfrak{J}) \\ &= d\ln(\|\mathbf{w}\|) + \frac{1}{2}\ln\frac{|\Delta_K|}{d^d} + d\ln\frac{\sqrt{d}\mathrm{N}(\mathbf{w})^{1/d}}{\|\mathbf{w}\|} + \ln \mathrm{N}(\mathfrak{J})\end{aligned}$$

To obtain a formula for the average-case behaviour, we take expectations and follow the heuristic analysis in [DEP25]. For large enough ranks $b \geq \lfloor 45/d \rfloor$,

$$\mathbb{E}[\ell'_i] \approx d\,\mathrm{lgh}_K(b) + \frac{1}{2}\ln\frac{|\Delta_K|}{d^d} + d\,\mathrm{skew}_K(b) + 0$$

For smaller ranks $b \leq \lfloor 45/d \rfloor$, the heuristic analysis is inaccurate, and we instead use experimental data for the average profile of a mHKZ$_K$-reduced rank $\lfloor 45/d \rfloor$ module lattice.

We iteratively apply this reasoning at each position in the basis, to obtain the simulator $\mathsf{mBKZSim}_K$ presented in Algorithm 7. We note that setting $K = \mathbb{Q}$ recovers the original simulator BKZSim.

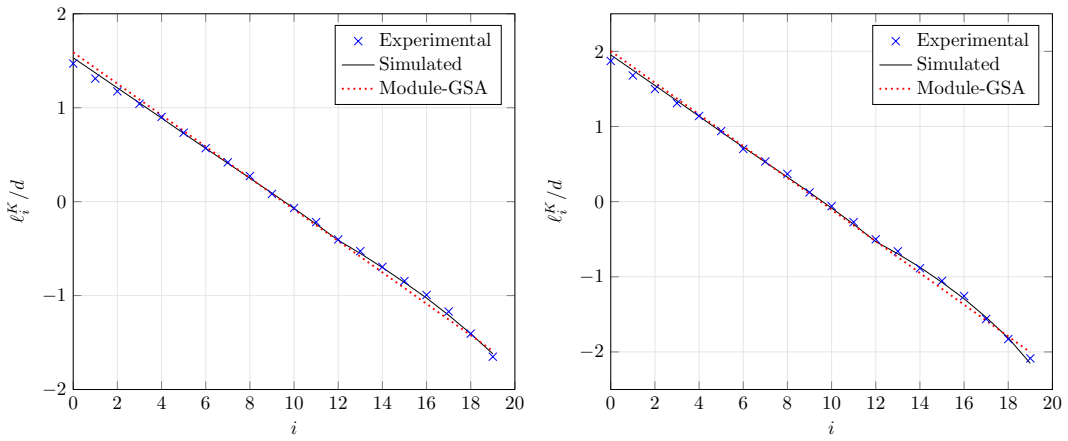
$\mathsf{mBKZSim}_K((\ell_i)_{i=1}^{\rho}, \beta_K, \tau)$

/* Pre-processing */
$(h_i) \leftarrow$ average profile of mHKZ$_K$-reduced unit-volume rank $\lfloor 45/d \rfloor$ module lattice
for $b \leftarrow 1$ **to** $\lfloor 45/d \rfloor$ **do**
  $c_b \leftarrow h_{\lfloor 45/d \rfloor - b + 1} - \frac{1}{b} \sum_{i=\lfloor 45/d \rfloor - b + 1}^{\lfloor 45/d \rfloor} h_i$
end
for $b \leftarrow \lfloor 45/d \rfloor + 1$ **to** β_K **do**
  $c_b \leftarrow d \lg \mathrm{h}_K(b) + \frac{1}{2} \ln \frac{|\Delta_K|}{d^d} + d \, \mathrm{skew}_K(b)$
end
/* Profile simulation */
for tour $\leftarrow 1$ **to** τ **do**
  $\phi \leftarrow$ true
  $t \leftarrow \min(\lfloor 45/d \rfloor, \beta_K)$
  for $i \leftarrow 1$ **to** $\rho - t + 1$ **do**
    $b \leftarrow \min(\beta_K, \rho - i)$ // Dimension of local block
    $f \leftarrow \min(i + \beta_K, \rho)$ // End index of local block
    $\ln V \leftarrow \sum_{j=1}^{f} \ell_j - \sum_{j=1}^{i-1} \ell'_j$
    if $\phi =$ true **then**
      if $\ln V/b + c_b < \ell_i$ **then**
        $\ell'_i \leftarrow \ln V/b + c_b$
        $\phi \leftarrow$ false
      end
    end
    else
      $\ell'_i \leftarrow \ln V/b + c_b$
    end
  end
  // Insert tail
  $\ln V \leftarrow \sum_{i=1}^{\rho} \ell_i - \sum_{i=1}^{\rho-t} \ell'_i$
  for $i \leftarrow \rho - t + 2$ **to** ρ **do**
    $\ell'_i \leftarrow \ln V/t + h_{i+\lfloor 45/d \rfloor - \rho}$
  end
  // Overwrite old profile with new profile
  $(\ell_i)_{i=1}^{\rho} \leftarrow (\ell'_i)_{i=1}^{\rho}$
end
return $(\ell_i)_{i=1}^{\rho}$

Algorithm 7: Proposed algorithm for simulating τ tours of mBKZ$_K^{\beta_K}$ on a given K-profile $(\ell_i)_{i=1}^{\rho}$.

4.2 Implementation

We provide a Python implementation of the simulator, which we use in Figure 6 to compare experimental and simulated K-profiles. We see in particular the tails of the simulated profiles reflect those of the experimental profiles more accurately than the simple Module-GSA.



(a) $c = 15$ **(b)** $c = 16$

Figure 6: Comparison between the Module-GSA, the simulated profiles using Algorithm 7, and the experimental profiles taken from [DEP25]. Each experimental profile is the average over 5 lattices. Random module lattices of $\mathbb{Q}$-rank 160 were reduced with blocksize $\beta_K = 64/d$ for 30 tours.

5 The Module Primal Attack

The standard primal attack can be applied to Module-LWE instances, but does not utilise the algebraic structure. On the other hand, Module-BKZ allows for a module analogue of the primal attack. We generalise our improved model for the primal attack to the module setting, and compare the two attacks.

Definition 26 (Module-Bai–Galbraith Embedding). *Given an instance of normal form Module-LWE,* $(\mathbf{A}, \mathbf{b}) \leftarrow \mathrm{MLWE}_{K,r_K,m_K,q,\chi}$, *let* $\rho_K = r_K + m_K + 1$. *We define the Module-Bai–Galbraith lattice as the the rank* ρ_K *free module* $\mathcal{M}$ *over* $\mathcal{O}_K$ *with* K*-basis given by the columns of*

$$\mathbf{M} = \begin{pmatrix} q\mathbf{I}_{m_K} & -\mathbf{A}^T & \mathbf{b} \\ & \mathbf{I}_{r_K} & \\ & & t \end{pmatrix} \in K^{\rho_K \times \rho_K}$$

where $t \in K$ *is the embedding coefficient.*

The module lattice $\mathcal{M}$ contains a rank 1 module sublattice $\mathcal{V} = \mathbf{v}\mathcal{O}_K$, where $\mathbf{v} = (\mathbf{e}, \mathbf{s}, t)^T$ contains the error $\mathbf{e}$ and secret $\mathbf{s}$. From the K-basis we may compute the volume $\det_{\mathbb{Q}}(\mathcal{M}) = |\Delta_K|^{\rho_K/2} q^{m_K d} \mathrm{N}(\|t\|_K)$. The *module primal attack* against MLWE consists of forming this lattice, and applying mBKZ with the goal of recovering $\mathcal{V}$.

The Shortest Vector. Unlike in the standard primal attack where the embedding gives a uSVP instance, we have instead embedded a dense sublattice $\mathcal{V} \subset \mathcal{M}$ which contains many short vectors $\{\zeta\mathbf{v}\}_{\zeta \in \mu_K}$. While $\mathbf{v} \in \mathcal{V}$ minimises the *algebraic norm*, it is possible that there

is some element in $\mathbf{w} \in \mathcal{V}$ with shorter Euclidean norm. However, we can justify that this is unlikely when ρ_K is sufficiently large. For any $\mathbf{w} \in \mathcal{V}$ we have

$$\|\mathbf{w}\| \geq \sqrt{d}\mathrm{N}(\mathbf{w})^{1/d} \geq \sqrt{d}\mathrm{N}(\mathbf{v})^{1/d}$$

and, as ρ_K increases, we expect this lower bound to rapidly tend to $\|\mathbf{v}\|$. Indeed, modelling $\mathbf{v}$ as spherically sampled,[13] we may apply the skewness model $\mathrm{skew}_K(\rho_K)$ (see §2.9) as a heuristic lower bound,

$$0 \geq \mathbb{E}_{\mathbf{v}}\left[\ln \frac{\sqrt{d}\mathrm{N}(\mathbf{v})^{1/d}}{\|\mathbf{v}\|}\right] \geq \mathrm{skew}_K(\rho_K) = \frac{1 - d_{\mathbb{R}} - d_{\mathbb{C}}}{2d\rho_K} + o\left(\frac{1}{\rho_K}\right) .$$

In particular, even if there *is* a shorter vector, it is not much shorter than $\mathbf{v}$. We henceforth always make the assumption that $\mathbf{v}$ is a shortest vector, and moreover that $\mu_K\mathbf{v} = \{\zeta\mathbf{v}\}_{\zeta\in\mu_K}$ is the set of shortest vectors. We note that a shorter vector $\mathbf{w} \in \mathcal{V}$ would increase the chance of mBKZ recovering $\mathcal{V}$ and hence $\mathbf{v}$, so this assumption is pessimistic for an attacker.

Choice of Embedding Coefficient. The notion of f-unusual-SVP does not demand uniqueness of the shortest vector, and we may still heuristically assume that the hardness of recovering any one of the shortest vectors is driven by the ratio $f = \mathrm{gh}_K(\mathcal{M})/\lambda_1(\mathcal{M})$. To maximise f, note the factor dependent on t is

$$\frac{\mathrm{N}(\|t\|_K)^{1/d(r_K+m_K+1)}}{\sqrt{d(r_K+m_K)\sigma^2 + \|t\|^2}}$$

and we recall the inequality $\mathrm{N}(\|t\|_K)^{1/d} \leq \|t\|/\sqrt{d}$, with equality when $t \in \mathbb{Q}\cdot\mu_K$. Thus we choose t rational; the above then simplifies, and is maximised at $t = \sigma$.[14]

5.1 Blocksize prediction using the Module-GSA

We now adapt the [ADPS16] blocksize analysis to the module setting, estimating the blocksize β_K required for the module primal attack to succeed, assuming the mGSA. Let $(\mathbf{A}, \mathbf{b}) \leftarrow \mathrm{MLWE}_{K,r_K,m_K,q,\chi}$, and let $\mathcal{M}$ be the corresponding Module-Bai–Galbraith lattice with embedded vector $\mathbf{v}$. When a tour of $\mathrm{mBKZ}_K^{\beta_K}$ reaches the terminal block, a projection of an embedded vector $\mathbf{w} \in \mu_K\mathbf{v}$ is inserted if it is a shortest vector. Under any K-linear projection π, the projections of the embedded vectors will all have the same length,

$$\|\pi(\zeta\mathbf{v})\| = \|\zeta\pi(\mathbf{v})\| = \|\pi(\mathbf{v})\| \quad \forall\zeta \in \mu_K$$

hence we need only consider one shortest vector $\mathbf{v}$, leading to the success condition

$$\|\pi_{\rho_K-\beta_K+1}(\mathbf{v})\| \leq \lambda_1(\mathcal{M}_{[\rho_K-\beta_K+1]}). \tag{4}$$

Using the beta distribution model from §3.1 for the squared norm of the projection of $\mathbf{v}$, we compute the expected logarithm of the norm as

$$\mathbb{E}\left[\ln\|\pi_{\rho_K-\beta_K+1}(\mathbf{v})\|\right] = \mathbb{E}\left[\ln\|\mathbf{v}\|\right] + \frac{1}{2}\,\mathbb{E}\left[\ln X\right]$$

where $X \sim \mathrm{Beta}\left(\frac{d\beta_K}{2}, \frac{d(\rho_K-\beta_K)}{2}\right)$. Following the [ADPS16] paradigm we consider the case where $\mathbf{v}$ is a spherical Gaussian, yielding

$$\mathbb{E}\left[\ln\|\pi_{\rho_K-\beta_K+1}(\mathbf{v})\|\right] = \ln(\sigma\sqrt{2}) \;+\; \frac{1}{2}\,\psi\left(\frac{d\beta_K}{2}\right) = \ln\left(\sigma\sqrt{d\beta_K}\right) - \frac{1}{2d\beta_K} + o\left(\frac{1}{\beta_K}\right).$$

[13] Such a distribution is obtained through a spherical sampling on the cyclic embedding.
[14] If σ is irrational, we may choose an arbitrarily good rational approximation.

On the other hand, after running $\mathrm{mBKZ}_K^{\beta_K}$, we assume the shortest vector in the terminal block behaves according to the module Gaussian heuristic,

$$\lambda_1(\mathcal{M}_{[\rho_K-\beta_K+1]}) = \mathrm{gh}_K(\beta_K) \cdot \det_{\mathbb{Q}}(\mathcal{M}_{[\rho_K-\beta_K+1]})^{\frac{1}{d\beta_K}}$$

From the definition of the K-profile and the mGSA we derive

$$\mathbb{E}\left[\ln \det_{\mathbb{Q}}(\mathcal{M}_{[\rho_K-\beta_K+1]})\right] = \beta_K\left(\frac{1}{\rho_K}\ln\det_{\mathbb{Q}}(\mathcal{M}) + \frac{\rho_K-\beta_K}{2}\mathrm{slope}_K(\beta_K)\right)$$

Note that $\mathrm{slope}_K(\beta_K) = -\ln\alpha_K < 0$ from Definition 25. Taking logs and expectations in Equation (4), substituting the above and simplifying, we attain the mGSA success condition:

$$\ln(\sigma\sqrt{2}) + \frac{1}{2}\psi\left(\frac{d\beta_K}{2}\right) < \mathrm{lgh}_K(\beta_K) + \frac{\ln\det_{\mathbb{Q}}(\mathcal{M})}{d\rho_K} + \frac{\rho_K-\beta_K}{2d}\mathrm{slope}_K(\beta_K) \tag{5}$$

5.2 Asymptotic comparison to the standard primal attack

The standard primal attack can also be applied to MLWE instances, by ignoring the module structure and embedding only one vector (in the language of Definition 27 later, using the ℓ-embedding with $\ell = 1$). The resulting lattice Λ has rank $d(r_K + m_K) + 1$, volume $|\Delta_K|^{d(r_K+m_K)/2}q^{m_K d}\sigma$, and follows the GSA success condition

$$\ln(\sigma\sqrt{2}) + \frac{1}{2}\psi\left(\frac{\beta}{2}\right) < \mathrm{lgh}_{\mathbb{Q}}(\beta) - \frac{d(r_K+m_K)+1-\beta}{\beta-1} + \frac{\mathrm{vol}(\Lambda)}{d(r_K+m_K)+1} \tag{6}$$

We may thus asymptotically compare the blocksizes required by the standard and module primal attacks.

Theorem 1 (Asymptotic Relation Between BKZ and Module-BKZ Blocksizes). *Let $\beta = d\beta_K$ where β_K is the minimal blocksize satisfying the Module-GSA success condition Equation (5). Let β_{eq} be the minimal blocksize satisfying the corresponding GSA success condition, Equation (6).*

If σ and q are fixed and $m_K = \alpha r_K + o(r_K)$, then

$$\beta_{\mathrm{eq}} = \beta + \frac{1}{d}\ln\frac{d^d}{|\Delta_K|}\ln\left(\frac{q^{2\alpha/(1+\alpha)}|\Delta_K|^{1/d}}{\sigma^2 2\pi e}\right)\frac{\beta}{(\ln\beta)^2} + o\left(\frac{\beta}{(\ln\beta)^2}\right).$$

Proof. See Appendix C.2.

To prove the above, we rely on the following lemma relating $r_K + m_K$ to β asymptotically, which is itself insightful.

Lemma 2 (Asymptotic Relation Between Dimension and Blocksize in the Module Primal Attack). *Let $r = dr_K$, $m = dm_K$, and let $\beta = d\beta_K$ where β_K is the minimal blocksize satisfying the Module-GSA success condition Equation (5). If σ and q are fixed and $m_K = \alpha r_K + o(r_K)$, then*

$$r + m = \beta + C_f \cdot \frac{\beta}{\ln\beta} + C_K C_f \cdot \frac{\beta}{(\ln\beta)^2} + o\left(\frac{\beta}{(\ln\beta)^2}\right),$$

where $C_f = \ln\left(\frac{q^{2\alpha/(1+\alpha)}|\Delta_K|^{1/d}}{\sigma^2 2\pi e}\right)$ *and* $C_K = \frac{1}{d}\ln\frac{d^d}{|\Delta_K|} + \ln(2\pi e)$.

Proof. See Appendix C.1.

When K is a cyclotomic number field with conductor a power of two, then $|\Delta_K| = d^d$ and thus Theorem 1 simply implies that $\beta_{\text{eq}} - \beta = o(\beta/(\ln \beta)^2)$. On the other hand, for all other cyclotomic fields, $|\Delta_K| < d^d$ and so the module primal attack succeeds with smaller blocksizes, with the gap driven by the factor $\ln(d^d/|\Delta_K|)/d$. This factor, corresponding to the 'discriminant gap' term in [DEP25], depends only on the radical (product of distinct prime factors) of the conductor, growing with the number of small odd primes in its decomposition [DEP25, §4.3].

The other factor C_f increasing the gap is directly impacted by the $\mathbb{Q}$-determinant of the module lattice and the length of its shortest vector – though related to usual measures of hardness of a short vector recovery, the importance of the discriminant in its value should be noted.

5.3 Prediction via Simulation

For more fine-grained predictions, we adapt our SVP estimators (Algorithms 5 and 6) to obtain $\mathsf{mBKZSuccess}$ and $\mathsf{mProgBKZSuccess}$. The first is presented in Algorithm 8, and the second follows analogously.

$\mathsf{mBKZSuccess}_K((\ell_i^K)_{i=1}^{\rho}, \lambda_1, \beta_K, \tau)$

$p_{\text{tot}} \leftarrow 0$
for `tour` $\leftarrow 1$ **to** τ **do**
  $(\ell_i)_{i=1}^{\rho} \leftarrow \mathsf{mBKZSim}((\ell_i^K)_{i=1}^{\rho}, \beta_K, 1)$
  $\ln V \leftarrow \sum_{i=\rho-\beta_K+1}^{\rho} \ell_i^K$ `// Volume of terminal block`
  $\lambda_1' \leftarrow \exp(\text{lgh}_K(\beta_K) + \ln V/d\beta_K)$ `// mGH for terminal block`
  $p_{\text{new}} \leftarrow \mathbb{P}\Big(\lambda_1^2 \cdot \text{Beta}\Big(\frac{d\cdot\beta_K}{2}, \frac{d\cdot\rho - d\cdot\beta_K}{2}\Big) \le (\lambda_1')^2\Big)$
  $p_{\text{tot}} \leftarrow p_{\text{tot}} + (1 - p_{\text{tot}}) \cdot p_{\text{new}}$
end
return p_{tot}

Algorithm 8: Proposed estimator for the success probability of solving SVP via τ tours of $\text{mBKZ}_K^{\beta_K}$.

To apply these to Module-LWE we additionally (as in §3.2) require an initial profile for the Module-Bai–Galbraith lattice, and the probability distribution of $\|\mathbf{v}\|^2$. For the initial profile, we use a simple adaptation of LLLSim returning instead a K-profile, which we call $\mathsf{mLLLSim}$. In our experiments, we use $\mathbf{v} \sim \chi_\sigma^{\rho_K}$, and thus model $\|\mathbf{v}\|^2 \sim \sigma^2 \chi^2_{d\rho_K}$.

5.4 Experiments

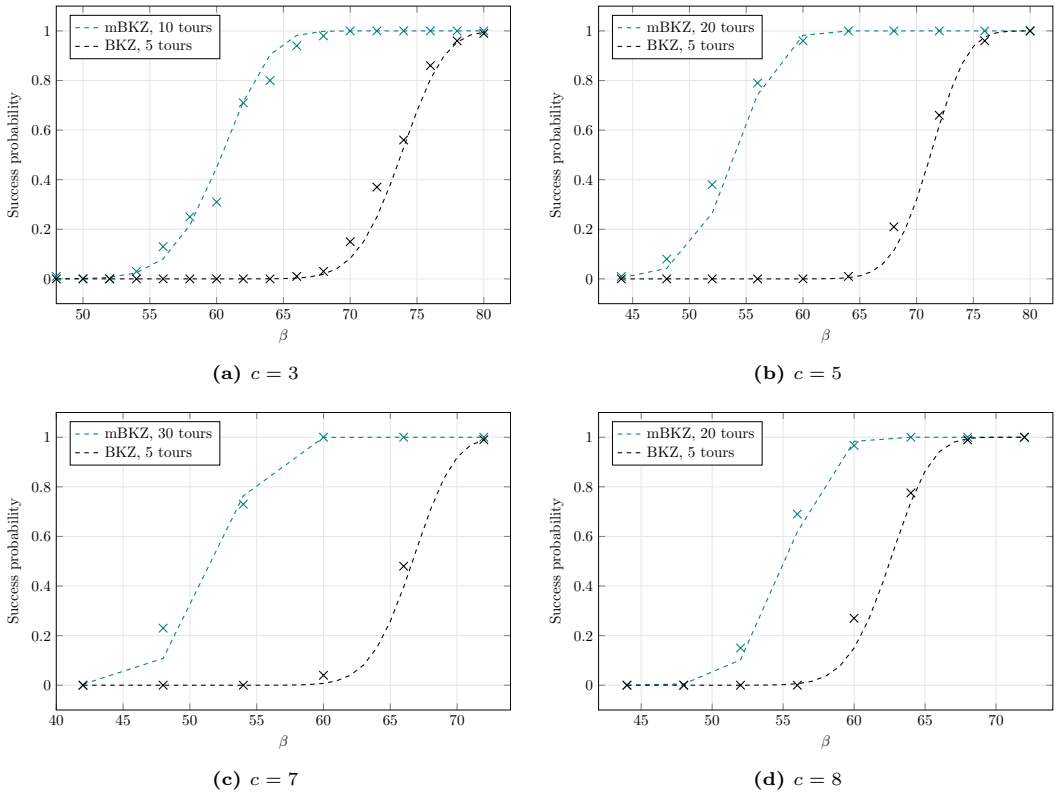
MLWE parameters. We consider Module-LWE with four parameter sets, listed in Table 2. Each parameter set consist of the conductor c for the underlying number field $K = \mathbb{Q}(\omega_c)$, as well as the MLWE parameters r_K, m_K, q, and a parameter σ for the width of $\chi = \chi_\sigma$ as per Definition 19. We also provide the GSA and mGSA predictions for the standard and module primal attacks, given as $\mathbb{Q}$-blocksizes β_{mGSA} and β_{GSA} respectively. The parameter sets are chosen to keep β_{mGSA} roughly constant while comparing different number fields.[15]

Implementation Details. We provide an implementation of the stuctured primal attack in Python, building upon the implementation of Module-BKZ provided in [DEP25]. Module lattices are represented using the cyclic embedding. The SVP oracle is implemented using sieving, as provided by (the Python wrapper for) the `g6k` library [ADH+19].

[15] Interestingly, for $c = 7$ we have $\beta_{\text{mGSA}} > \beta_{\text{GSA}}$ despite mBKZ providing an improved slope; this is due to the differences in the rank, volume, and shortest vector length in the embedding lattices.

c	r_K	m_K	q	σ	β_{mGSA}	β_{GSA}
3	59	60	48000	11	60	66
5	29	30	30000	12	60	62
7	18	21	10000	10	60	56
8	29	30	20000	8	60	49

Table 2: MLWE parameter sets used for testing the primal attacks. We choose a fixed number of samples m_K slightly larger than r_K, as is typically optimal, see [ADPS16, Table 1].



(a) $c = 3$ **(b)** $c = 5$

(c) $c = 7$ **(d)** $c = 8$

Figure 7: Success probabilities for both the module primal attack using $\text{mBKZ}_K^{\beta/d}$, and the standard primal attack using BKZ-β. Dashed lines are estimations, crosses are experimental data. For each blocksize in each subfigure, 200 experiments were run. See Table 2 for full parameter sets.

Methodology for Comparison. In our experiments, we allow fair comparison between BKZ and Module-BKZ as follows.

For primal attacks using a fixed blocksize, we compare BKZ-β to $\text{mBKZ}_K^{\beta_K}$, where $\beta = d \cdot \beta_K$ is the $\mathbb{Q}$-blocksize of $\text{mBKZ}_K^{\beta_K}$, ensuring the SVP oracles in each algorithm operate on lattices of the same dimension. Furthermore, a single tour of BKZ applies the oracle at each consecutive index i, whereas in mBKZ it is only applied at indices which are a multiple of d. Thus to balance the number of oracle calls, we compare τ tours of BKZ to $d \cdot \tau$ tours of mBKZ.

For primal attacks using a progressive strategy, the situation is more complicated. Progressive-BKZ uses each blocksize $\beta \geq 2$, whereas Progressive-mBKZ can only utilise $\mathbb{Q}$-blocksizes $\beta \geq 2d$ that are multiples of d. Nevertheless, we will compare Progressive-BKZ with τ tours at each blocksize to Progressive-mBKZ with $d \cdot \tau$ tours at each blocksize, which

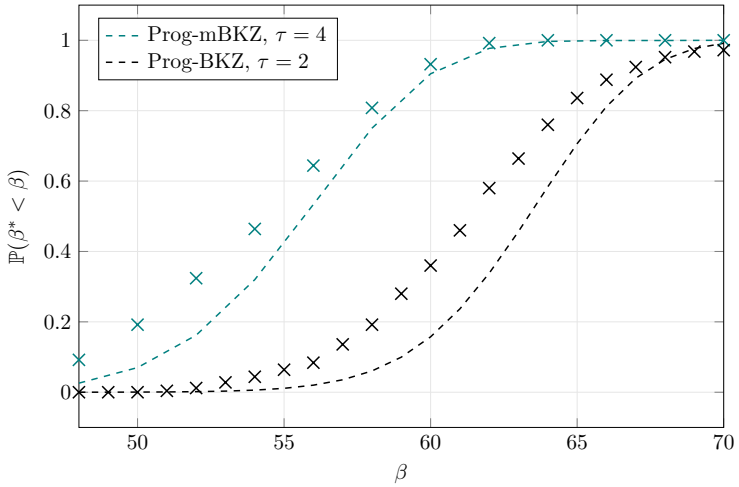


Figure 8: Success probabilities for both the module primal attack using Progressive-mBKZ$_K$ and the standard primal attack using Progressive-BKZ, applied to Module-LWE instances with the $c = 3$ parameter set (Table 2). Dashed lines are estimations, crosses are experimental data. For each attack, 250 experiments were run.

ensures close runtimes while keeping the cost for Progressive-mBKZ slightly lower – a more detailed cost comparison may be found in Appendix D.

Results & Observations In Figures 7 and 8 we compare the estimators to experiments. We observe that our estimators are relatively accurate, although tend to underestimate the success probability, especially in Figure 8. We suspect this is a consequence of optimisations in `g6k`, especially on-the-fly lifting, which are not captured in the model.

Regardless, the experiments and predictions show that the module primal attack appears superior in all cases, even when $\beta_{\text{mGSA}} > \beta_{\text{GSA}}$. To explain this, note that the GSA and mGSA success conditions assume sufficiently many tours have been run for convergence, and thus the relative convergence speeds of the algorithms are unaccounted for. For example in the $c = 8$ parameter set, writing τ and $\tau_K = d \cdot \tau$ for the number of tours in each case, the estimators predict that the standard attack is eventually superior when $\tau \geq 11$.

Furthermore, note that when balancing total oracle calls as we have done, that Module-BKZ makes a factor d *more* oracle calls on the terminal block ($d\tau$ as opposed to τ), and thus has more 'chances' to find a projection of the embedded vector, which is not captured in the GSA/mGSA.

6 Modelling attacks with multiple embedded vectors

Thus far, we have considered two attacks against Module-LWE: the standard primal attack, which embeds one short vector to give a uSVP instance; and the module primal attack, which embeds d short vectors generating a dense sublattice. To use Module-BKZ, it is necessary to embed the entire dense sublattice. However if we only consider BKZ, we are free to embed only part of it, generated by $1 \leq \ell \leq d$ short vectors.

It is possible that choosing $\ell > 1$ may be advantageous compared to the usual primal attack using $\ell = 1$. Intuitively, embedding more vectors should increase the probability at least one projection is short.[16]. Embeddings of this form have previously been considered

[16] Recall that Module-BKZ uses K-linear projections, and thus all embedded vectors had the same length after projection into the terminal block. On the contrary, BKZ uses only $\mathbb{Q}$-linear projections, allowing the projected lengths to differ.

in [NY21, UOM24], where it was observed experimentally for certain parameter sets that $\ell \approx 2, 3$ was optimal. In this section we investigate such embeddings, proposing a model for the success probability which takes into account the presence of multiple embedded vectors.

6.1 Extended Bai–Galbraith Embeddings

Let K be a cyclotomic field with $\theta : K \to \mathcal{C} \cong \mathbb{R}^c$ the cyclic embedding.

Definition 27 (ℓ-embedding). *For $(\mathbf{A}, \mathbf{b}) \leftarrow \mathrm{MLWE}_{K,r,m,q,\chi}$ and an integer $1 \le \ell \le d$, we define the ℓ-embedding as the matrix*

$$\mathbf{M}_\ell = \left[\begin{array}{c|c} \mathbf{B} & \begin{matrix}\uparrow & & \uparrow \\ \mathbf{b}_0 & \cdots & \mathbf{b}_{\ell-1} \\ \downarrow & & \downarrow\end{matrix} \\ \hline & \\ & \mathbf{T}_\ell \\ & \end{array}\right] \in \mathbb{R}^{(cm+cr+\ell)\times(dm+dr+\ell)}$$

where $\mathbf{B} \in \mathbb{R}^{c(m+r)\times d(m+r)}$ is a $\mathbb{Z}$-basis, under the cyclic embedding θ, for the module lattice with K-basis

$$\begin{bmatrix} q\mathbf{I}_m & -\mathbf{A}^T \\ & \mathbf{I}_r \end{bmatrix},$$

the vectors $\mathbf{b}_i = \theta(\omega_c^i \mathbf{b}) \in \mathbb{R}^{c(m+r)}$ are rotations of $\mathbf{b}$ in the cyclic embedding, and $\mathbf{T}_\ell \in \mathbb{R}^{\ell\times\ell}$ is the 'embedding submatrix' with columns $\mathbf{t}_i$.

This yields a (non full rank) lattice $\Lambda = \Lambda(\mathbf{M}_\ell) \subset \mathbb{R}^{cm+cr+\ell}$ of rank $N = dm+dr+\ell$, which contains the short vectors $\mathbf{v}_i = \big(\theta(\omega_c^i \mathbf{e}), \theta(\omega_c^i \mathbf{s}), \mathbf{t}_i\big)$ for $i = 0, \ldots, \ell-1$. Furthermore, suppose there is another root of unity in the $\mathbb{Z}$-span of the first ℓ, say $\omega_c^k = \sum_{i=0}^{\ell-1} z_i \omega_c^i$. Then for each such ω_c^k (up to sign), Λ additionally contains a short vector $\mathbf{v}_k = \Big(\theta(\omega_c^k \mathbf{e}), \theta(\omega_c^k \mathbf{s}), \sum_{i=0}^{\ell-1} z_i \mathbf{t}_i\Big)$.

6.2 Success Probability Model

To model the success probability of recovering an embedded vector with BKZ, we adapt our existing estimators (Algorithms 5 and 6) as follows.

First, we compute the set R of all roots of unity in $\mathrm{span}_{\mathbb{Z}}\{1, \omega_c, \ldots \omega_c^{\ell-1}\}$, up to sign, represented as coefficients $\mathbf{z} = (z_0, \ldots, z_{\ell-1})$ of the linear combination.[17]

Then, given $V = \|(\mathbf{e}, \mathbf{s})\|$, the corresponding vectors in Λ have square norms $V^2 + \|\sum_i z_i \mathbf{t}_i\|^2$. We make the assumption that their projections into the terminal block are independent. This heuristic has previously been considered, for example in [BN24, Heuristic 1]. Thus the shortest projection has square norm distributed as the minimum M of $|R|$ beta distributions,

$$M \sim \min_{\mathbf{z}\in R} \left(\left(V^2 + \left\| \sum_i z_i \mathbf{t}_i \right\|^2 \right) \cdot \mathrm{Beta}\left(\frac{\beta}{2}, \frac{N-\beta}{2} \right) \right).$$

So after each oracle call in the terminal block, the smallest projection is inserted with probability $\mathbb{P}(M \le \exp(2 \cdot \ell_{N-\beta+1}))$. By simulating the profile with BKZSim, we attain a success probability prediction for each V. To apply this to LWE, we integrate over the distribution of V (see §3.2).[18]

[17] This depends only on the conductor, and can be precomputed with some basic number theory.
[18] For our initial profile, we use a Z-shape prediction, adapted for ℓ-embeddings.

6.3 Experiments and Observations

We provide implementations of the ℓ-embedding attacks, and corresponding estimators, in our codebase. For the embedding submatrix we choose $\mathbf{T}_\ell = (\sigma\sqrt{\ell})\mathbf{I}_\ell$, which we justify in Appendix E. In Figure 9 we compare our estimations and experiments for a fixed blocksize, number of tours, and Module-LWE parametersiation, with ℓ varying. Additionally, we show the 'naive prediction' when the presence of multiple embedded vectors is not accounted for.

We observe that our model improves upon the naive prediction, but still underestimates success probabilities; a more thorough analysis of these ℓ-embeddings, and in particular of the angle dependency between the shortest vectors (and their projections) should be considered for future work.

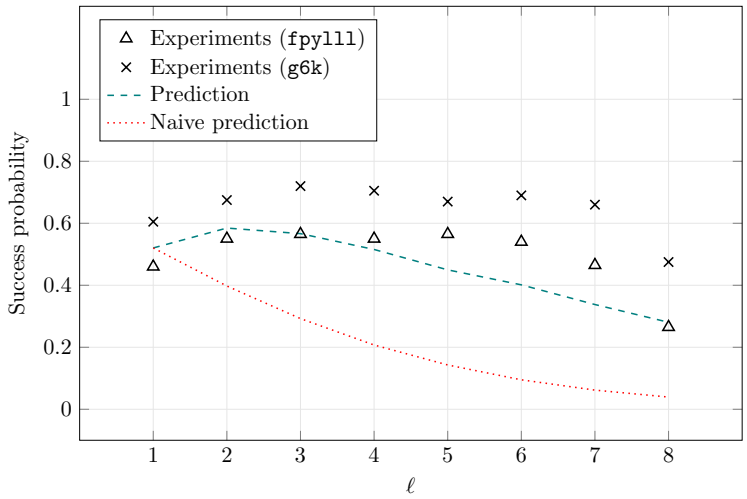


Figure 9: Comparison of predicted and experimental success probabilities for the ℓ-embedding primal attacks. We use the Module-LWE parameters $c = 15$, $r = 14$, $m = 15$, $q = 2100$, $\sigma = 4$ and run BKZ-58 for 10 tours. For each ℓ, 200 experiments were run, using both `fpylll` (enumeration) and `g6k` (sieving).

Acknowledgements. We thank Léo Ducas and Eamonn Postlethwaite for helpful discussions and comments, as well as King's College London e-Research [dt25b] for providing the computational resources used to run our experiments. Paola de Perthuis was supported by the NWO Gravitation Project QSC.

References

ACD+18. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 351–367. Springer, Cham, September 2018.

ACPS09. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Berlin, Heidelberg, August 2009.

AD21. Martin Albrecht and Léo Ducas. Lattice attacks on NTRU and LWE: A history of refinements. Cryptology ePrint Archive, Report 2021/799, 2021.

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, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 717–746. Springer, Cham, May 2019.

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.

AGPS20. Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck. Estimating quantum speedups for lattice sieves. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 583–613. Springer, Cham, December 2020.

AGVW17. Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 297–322. Springer, Cham, December 2017.

APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Math. Cryptol.*, 9(3):169–203, 2015.

Bai92. Ralph W. Bailey. Distributional identities of beta and chi-squared variates: A geometrical interpretation. *The American Statistician*, 46(2):117–120, 1992.

BDF18. Guillaume Bonnoron, Léo Ducas, and Max Fillinger. Large FHE gates from tensored homomorphic accumulator. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 18*, volume 10831 of *LNCS*, pages 217–251. Springer, Cham, May 2018.

BDGL16. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *27th SODA*, pages 10–24. ACM-SIAM, January 2016.

BDK+18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367. IEEE Computer Society Press, April 2018.

BG14. Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In Willy Susilo and Yi Mu, editors, *ACISP 14*, volume 8544 of *LNCS*, pages 322–337. Springer, Cham, July 2014.

BGV12. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 309–325, New York, NY, USA, 2012. Association for Computing Machinery.

BJRW23. Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. On the hardness of module learning with errors with short distributions. *Journal of Cryptology*, 36(1):1, January 2023.

BJTW25. Katharina Boudgoust, Corentin Jeudy, Erkan Tairi, and Weiqiang Wen. Hardness of m-LWE with general distributions and applications to leaky variants. Cryptology ePrint Archive, Paper 2025/1472, 2025.

BMW19. Shi Bai, Shaun Miller, and Weiqiang Wen. A refined analysis of the cost for solving LWE via uSVP. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje eddine Rachidi, editors, *AFRICACRYPT 19*, volume 11627 of *LNCS*, pages 181–205. Springer, Cham, July 2019.

BN24. Henry Bambury and Phong Q. Nguyen. Improved provable reduction of NTRU and hypercubic lattices. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part I*, pages 343–370. Springer, Cham, June 2024.

BSW18. Shi Bai, Damien Stehlé, and Weiqiang Wen. Measuring, simulating and exploiting the head concavity phenomenon in BKZ. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 369–404. Springer, Cham, December 2018.

CCHY24. Jung Hee Cheon, Hyeongmin Choe, Dongyeon Hong, and MinJune Yi. SMAUG: Pushing lattice-based key encapsulation mechanisms to the limits. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *SAC 2023*, volume 14201 of *LNCS*, pages 127–146. Springer, Cham, August 2024.

CN11. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, December 2011.

DD12. Léo Ducas and Alain Durmus. Ring-LWE in polynomial rings. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 34–51. Springer, Berlin, Heidelberg, May 2012.

DDGR20. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 329–358. Springer, Cham, August 2020.

DEP25. Léo Ducas, Lynn Engelberts, and Paola de Perthuis. Predicting module lattice reduction. *ASIACRYPT*, December 2025.

DP16. Leo Ducas and Thomas Prest. Fast fourier orthogonalization. pages 191–198, 07 2016.

DP26. Léo Ducas and Ludo N. Pulles. Accurate score prediction for dual-sieve attacks. *J. Cryptol.*, 39(1):8, 2026.

dt25a. The FPLLL development team. fpylll, a Python wrapper for the fplll lattice reduction library, Version: 0.6.4. Available at `https://github.com/fplll/fpylll`, 2025.

dt25b. The King's College London CREATE development team. King's Computational Research, Engineering and Technology Environment (CREATE). 2025.

DvW22. Léo Ducas and Wessel P. J. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 643–673. Springer, Cham, May / June 2022.

GSVV24. Nihar Gargava, Vlad Serban, Maryna Viazovska, and Ilaria Viglino. Effective module lattices and their shortest vectors. 2024.

How07. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 150–169. Springer, Berlin, Heidelberg, August 2007.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *LNCS*, pages 267–288. Springer, June 1998.

Kan87. Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

KK24. Alexander Karenin and Elena Kirshanova. Finding dense submodules with algebraic lattice reduction. In Serge Vaudenay and Christophe Petit, editors, *AFRICACRYPT 24*, volume 14861 of *LNCS*, pages 403–427. Springer, Cham, July 2024.

KP22. Jonghyun Kim and Jong Hwan Park. NTRU+: Compact construction of NTRU using simple encoding method. Cryptology ePrint Archive, Report 2022/1664, 2022.

LLL82. A. K. Lenstra, Jr. Lenstra, H. W., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

LM09. Vadim Lyubashevsky and Daniele Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 577–594. Springer, Berlin, Heidelberg, August 2009.

LMvdP13. Thijs Laarhoven, Michele Mosca, and Joop van de Pol. Solving the shortest vector problem in lattices faster using quantum search. In Philippe Gaborit, editor, *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013*, pages 83–101. Springer, Berlin, Heidelberg, June 2013.

LN20. Jianwei Li and Phong Q. Nguyen. A complete analysis of the BKZ lattice reduction algorithm. Cryptology ePrint Archive, Report 2020/1237, 2020.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010.

LPR13. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Berlin, Heidelberg, May 2013.

LPSW19. Changmin Lee, Alice Pellet-Mary, Damien Stehlé, and Alexandre Wallet. An LLL algorithm for module lattices. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 59–90. Springer, Cham, December 2019.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015.

MR07. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

MS20. Tamalika Mukherjee and Noah Stephens-Davidowitz. Lattice reduction for modules, or how to reduce ModuleSVP to ModuleSVP. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 213–242. Springer, Cham, August 2020.

NY21. Satoshi Nakamura and Masaya Yasuda. An extension of kannan's embedding for solving ring-based LWE problems. In Maura B. Paterson, editor, *18th IMA International Conference on Cryptography and Coding*, volume 13129 of *LNCS*, pages 201–219. Springer, Cham, December 2021.

PV21. Eamonn W. Postlethwaite and Fernando Virdia. On the success probability of solving unique SVP via BKZ. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 68–98. Springer, Cham, May 2021.

Reg09. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), September 2009.

Sch87. C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, 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.

SD17. Noah Stephens-Davidowitz. On the gaussian measure over lattices. 2017.

SE94. Claus Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 08 1994.

UOM24. Satoshi Uesugi, Shinya Okumura, and Atsuko Miyaji. Revisiting an Extension of Kannan's embedding for Ring-LWE. In Ilsun You, Michał Choraś, Seonghan Shin, Hwankuk Kim, and Philip Virgil Astillo, editors, *Mobile Internet Security*, pages 167–180, Singapore, 2024. Springer Nature Singapore.

A On the Beta Distribution Model

A.1 Proof of Lemma 1

Without loss of generality, we may scale to $\|\mathbf{v}\| = 1$. The idea is to 'rotate' the setup, so that the subspace is fixed and the vector is random.

Formally, let $S' = \text{span}\{\mathbf{e}_1, \ldots, \mathbf{e}_\beta\}$ be the span of the first β canonical basis vectors. By definition of a random subspace, $S = T(S')$ where $T \in O(N)$ is a uniformly random orthogonal transformation. Since lengths are preserved under orthogonal transformations, and the projections are orthogonal with respect to the inner product, we have $\|\pi_S(\mathbf{v})\| = \|\pi_{S'}(T^{-1}\mathbf{v})\|$.

Now $\mathbf{v}' = T^{-1}\mathbf{v}$ is distributed as a uniformly random vector on the sphere $S_{N-1} \subset \mathbb{R}^N$, or equivalently

$$\mathbf{v}' = \frac{(Z_1, \ldots, Z_N)}{\sqrt{Z_1^2 + \ldots + Z_N^2}}$$

where Z_i are iid. $\mathcal{N}(0,1)$ random variables. Thus,

$$\|\pi_S(\mathbf{v})\|^2 = \|\pi_{S'}(\mathbf{v}')\|^2 = \frac{Z_1^2 + \ldots + Z_\beta^2}{Z_1^2 + \ldots + Z_N^2} = \frac{X}{X+Y}$$

where $X = \sum_{i=1}^{\beta} Z_i^2$ and $Y = \sum_{i=\beta+1}^{N} Z_i^2$. Note that $X \sim \chi^2_\beta$, $Y \sim \chi^2_{N-\beta}$, and that X and Y are independent. The result follows by a standard statistical identity for the ratio $X/(X+Y)$, see for example [Bai92]. □

A.2 Square Norms of Discrete Gaussian Vectors

For $\mathbf{v} \sim D_{\mathbb{Z}^n,\sigma}$, we wish to compare the distribution of $\|\mathbf{v}\|^2$ to the continuous approximation given by the probability density function of the scaled chi-squared distribution $\sigma^2\chi^2_n$,

$$f_{n,\sigma}(v) = \frac{v^{\frac{n}{2}-1} e^{-\frac{v}{2\sigma^2}}}{(2\sigma^2)^{\frac{n}{2}}\Gamma\left(\frac{n}{2}\right)}.$$

Assuming v is sufficiently large, the Gaussian heuristic for $\mathbb{Z}^n$ states that the number of points in $B_n(v)$ and $B_n(v+1)$ can be approximated by the corresponding volumes, thus

$$\left|\left\{\mathbf{x} \in \mathbb{Z}^n, \|\mathbf{x}\|^2 \le v\right\}\right| \approx \frac{(v\pi)^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)}$$
$$\left|\left\{\mathbf{x} \in \mathbb{Z}^n, v < \|\mathbf{x}\|^2 \le v+1\right\}\right| \approx \frac{\pi^{\frac{n}{2}}\left((v+1)^{n/2} - v^{n/2}\right)}{\Gamma\left(\frac{n}{2}+1\right)}$$

Lemma 3. *Let $\mathbf{v} \sim D_{\mathbb{Z}^n,\sigma}$. Defining $b_\sigma = 1 + (2 + 1/(\sigma\sqrt{2\pi}))e^{-1/(2\sigma^2)}$ and $B_\sigma = 1 + 2e^{-1/(2\sigma^2)}$, we have the bounds*

$$\frac{1}{(\sigma\sqrt{2\pi})^n b_\sigma^n} \le \mathbb{P}\left[\|\mathbf{v}\|^2 = 0\right] \le \frac{1}{(\sigma\sqrt{2\pi})^n B_\sigma^n},$$

and, under the Gaussian heuristic for $\mathbb{Z}^n$ lattice, for each $v \in \mathbb{Z}_{>0}$,

$$\frac{A(v) f_{n,\sigma}(v)}{b_\sigma^n} \le \mathbb{P}\left[\|\mathbf{v}\|^2 = v\right] \le \frac{A(v) f_{n,\sigma}(v)}{B_\sigma^n}$$

where $A(v) = \frac{2v(1-(1-\frac{1}{v})^{n/2})}{n} = 1 + o_{v\to+\infty}(1)$.

Proof. Write $\rho(\mathbf{x}) = e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}}$ and $\rho(\mathbb{Z}^n) = \sum_{\mathbf{x}\in\mathbb{Z}^n} \rho(\mathbf{x})$. For each integer $v \geq 0$,

$$\mathbb{P}\left[\|\mathbf{v}\|^2 = v\right] = \sum_{\mathbf{x}\in\mathbb{Z}^n \,|\, \|\mathbf{x}\|^2 = v} \frac{e^{-\frac{v}{2\sigma^2}}}{\rho(\mathbb{Z}^n)}$$

We obtain bounds on $\rho(\mathbb{Z}^n)$ from [SD17, Claim 2.8.1],

$$\frac{(\sigma\sqrt{2\pi})^{-n}}{b_\sigma^n} \leq \frac{1}{\rho\left(\mathbb{Z}^n\right)} \leq \frac{(\sigma\sqrt{2\pi})^{-n}}{B_\sigma^n}$$

For the unique point with zero norm, the result follows. For $v > 0$, we approximate $\left|\left\{\mathbf{x} \in \mathbb{Z}^n, \|\mathbf{x}\|^2 = v\right\}\right|$ with the Gaussian heuristic as the volume of $B_n(v+1) \setminus B_n(v)$, giving

$$\frac{A(v) f_{n,\sigma}(v)}{b_\sigma^n} \leq \mathbb{P}\left[\|\mathbf{v}\|^2 = v\right] \leq \frac{A(v) f_{n,\sigma}(v)}{B_\sigma^n}$$

where $A(v) = \frac{\Gamma\left(\frac{n}{2}\right)\left(v^{\frac{n}{2}} - (v-1)^{\frac{n}{2}}\right)}{\Gamma\left(\frac{n}{2}+1\right) z^{\frac{n}{2}-1}}$, and as $\Gamma\left(\frac{n}{2}+1\right) = \frac{n}{2}\Gamma\left(\frac{n}{2}\right)$,

$$A(v) = \frac{2v(1-(1-\frac{1}{v})^{n/2})}{n} = 1 + o_{v\to+\infty}\left(1\right).$$

B Statistical Measures of Closeness

B.1 Total Squared Error

In Figure 3, the total squared error between estimated and experimental probabilities is compared to a statistical baseline: the expected total square error if the same number of samples were drawn again from the experimentally obtained distribution and used to estimate it. Here, we derive a formula for this baseline.

Considering Progressive-BKZ, write $f : i \mapsto p_i$ for the experimental probability mass function. Suppose we draw n independent samples from this distribution, $X_1, \ldots, X_n \sim f$, and then estimate the mass function by $\hat{p}_i = N_i/n$, where $N_i = \#\{X_j \mid X_j = j\} \sim \mathrm{Bin}(n, p_i)$, with $\mathbb{V}(N_i) = np_i(1-p_i)$. The expected total squared error is, using linearity of expectation,

$$\mathbb{E}\left[\sum_i (\hat{p}_i - p_i)\right] = \sum_i \mathbb{V}(\hat{p}_i) = \frac{1}{n^2}\sum_i \mathbb{V}(N_i) = \frac{p_i(1-p_i)}{n} \tag{7}$$

In the case of BKZ-β, ones similarly arrives at the same formula (where p_i is instead the experimental success probability when using blocksize $i = \beta$).

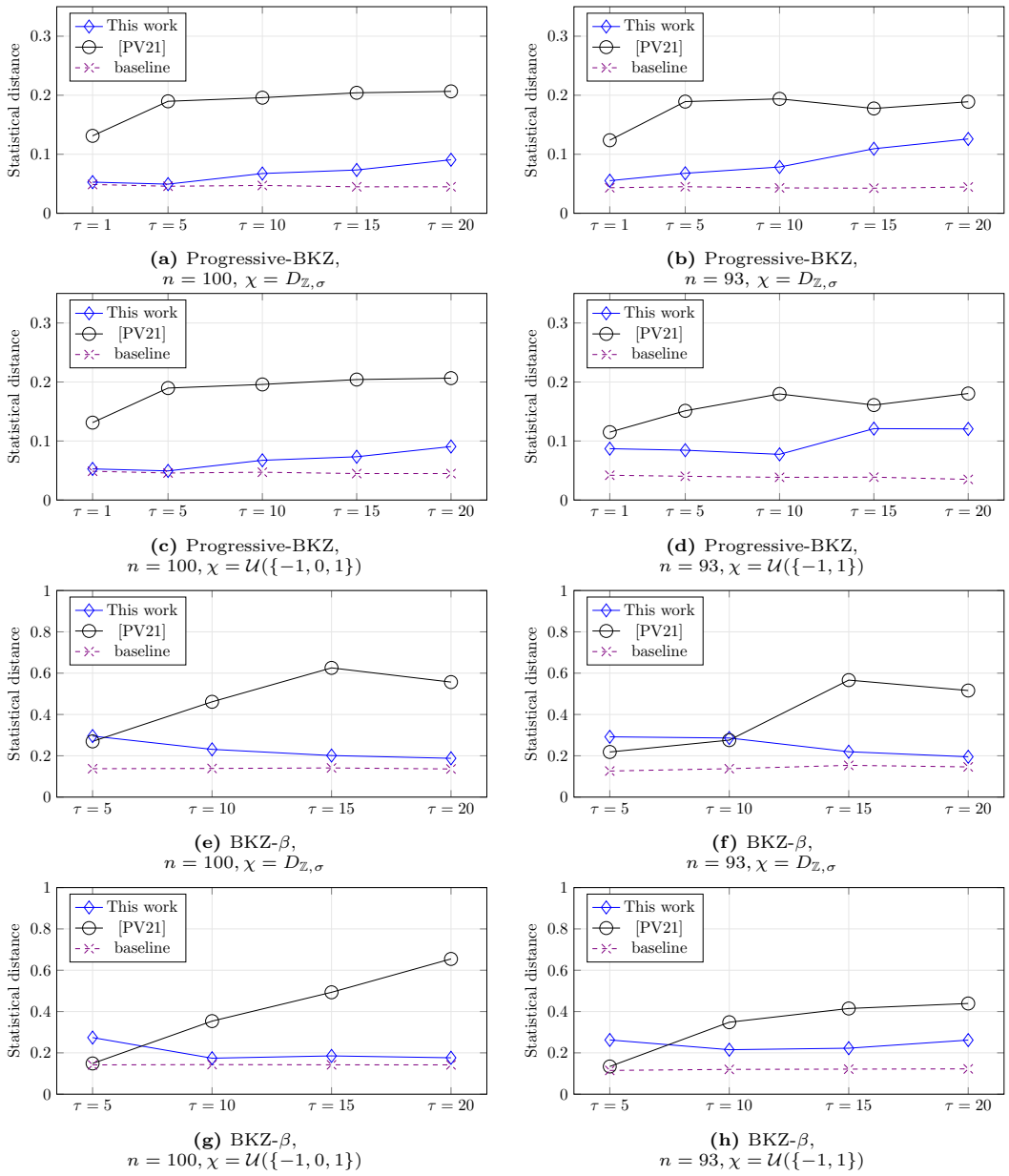
B.2 Statistical Distance

An alternative measure of closeness is the statistical distance, which is commonly used in cryptology, and which we use as a baseline in Figure 10.

In the case of Progressive-BKZ, we have experimental and estimated probability mass functions, f_{exp} and f_{est} respectively. The statistical distance is simply the statistical distance between the two corresponding distributions, $\Delta(f_{\mathrm{exp}}, f_{\mathrm{est}}) = \frac{1}{2}\sum_\beta |f_{\mathrm{exp}}(\beta) - f_{\mathrm{est}}(\beta)|$.

On the other hand, for BKZ-β, we obtain for each β an estimate $f_{\mathrm{est}}(\beta)$ of the success probability when using β, and an experimental probability $f_{\mathrm{exp}}(\beta)$. While these functions do *not* correspond to probability distributions, we nevertheless define an analogous statistical distance $\Delta(f_{\mathrm{exp}}, f_{\mathrm{est}}) = \frac{1}{2}\sum_\beta |f_{\mathrm{exp}}(\beta) - f_{\mathrm{est}}(\beta)|$.

In Figure 10, we compare our estimators with those from [PV21], showing the statistical distances between estimated and experimental success probabilities.



(a) Progressive-BKZ, $n = 100$, $\chi = D_{\mathbb{Z},\sigma}$

(b) Progressive-BKZ, $n = 93$, $\chi = D_{\mathbb{Z},\sigma}$

(c) Progressive-BKZ, $n = 100, \chi = \mathcal{U}(\{-1, 0, 1\})$

(d) Progressive-BKZ, $n = 93, \chi = \mathcal{U}(\{-1, 1\})$

(e) BKZ-β, $n = 100, \chi = D_{\mathbb{Z},\sigma}$

(f) BKZ-β, $n = 93, \chi = D_{\mathbb{Z},\sigma}$

(g) BKZ-β, $n = 100, \chi = \mathcal{U}(\{-1, 0, 1\})$

(h) BKZ-β, $n = 93, \chi = \mathcal{U}(\{-1, 1\})$

Figure 10: Statistical distance between estimated and experimental success probabilities, compared to a baseline of the expected statistical distance when sampling n times from the original experimental distribution, as explained in Appendix B.2.

Baseline. Again, we provide a statistical baseline: the expected statistical distance if the same number of samples were drawn again from the experimentally obtained distribution, and used to estimate it.

For Progressive-BKZ, resuming the notation of Equation (7), we compute this as

$$\mathbb{E}\left[\frac{1}{2}\sum_i \left|\frac{N_i}{n} - p_i\right|\right] = \frac{1}{2}\sum_i \left(E_i^+ + E_i^-\right)$$

where

$$E_i^+ = \sum_{j=\lfloor np_i \rfloor+1}^{n} \left(\frac{j}{n} - p_i\right)\binom{n}{j} p_i^j (1-p_i)^{n-j}$$
$$E_i^- = \sum_{j=0}^{\lfloor np_i \rfloor} \left(p_i - \frac{j}{n}\right)\binom{n}{j} p_i^j (1-p_i)^{n-j}$$

Using the identity $j\binom{n}{j} = n\binom{n-1}{j-1}$, and writing I for the regularized incomplete beta function, we obtain the formulae

$$\begin{aligned} E_i^- &= p_i \sum_{j=0}^{\lfloor np_i \rfloor} \binom{n}{j} p_i^j (1-p_i)^{n-j} - p_i \sum_{j=0}^{\lfloor np_i \rfloor - 1} \binom{n-1}{j} p_i^j (1-p_i)^{n-1-j} \\ &= p_i(I_{1-p_i}(n - \lfloor np_i \rfloor, 1 + \lfloor np_i \rfloor) - I_{1-p_i}(n - \lfloor np_i \rfloor, \lfloor np_i \rfloor)) \end{aligned}$$

and, likewise,

$$E_+ = p_i(I_{1-p_i}(n - \lfloor np_i \rfloor, 1 + \lfloor np_i \rfloor) - I_{1-p_i}(n - \lfloor np_i \rfloor, \lfloor np_i \rfloor)).$$

For BKZ, we consider for each $i > 0$ a binomial variable $X_i \sim \mathrm{Bin}(n, p_i)$ where n is the number of samples, and p_i the experimental probability using blocksize $\beta = i$. Our baseline expected distance will then be:

$$\mathbb{E}\left[\frac{1}{2}\sum_{i=1}^{B}\left|\frac{X_i}{n} - p_i\right|\right]$$

which leads to exactly the same formulae as for Progressive-BKZ.

C Asymptotic Analyses

C.1 Proof of Lemma 2

Let β_K be the first blocksize satisfying the Module-GSA success condition Equation (5), and write $r = dr_K$, $m = dm_K$, with $m = \alpha r + o(r)$. The Module-Bai–Galbraith lattice $\mathcal{M}$ (with embedding coefficient $t = \sigma$) has log-volume

$$\ln V_K = \ln \det_{\mathbb{Q}}(\mathcal{M}) = \frac{r+m+d}{2d}\ln|\Delta_K| + m\ln q + d\ln\sigma\ .$$

Considering asymptotics in Equation (5), as β grows,

$$\begin{aligned} \ln(\sigma\sqrt{\beta}) = \mathrm{lgh}_{\mathbb{Q}}(\beta) + o\left(\frac{1}{\ln\beta}\right) \\ - \frac{r+m+d-\beta}{\beta-d}\left(\mathrm{lgh}_{\mathbb{Q}}(\beta) + \frac{1}{2d}\ln\frac{|\Delta_K|}{d^d} + o(1)\right) + \frac{\ln V_K}{r+m+d} \end{aligned} \tag{8}$$

where one uses $\mathrm{lgh}_K(\beta_K) = \mathrm{lgh}_{\mathbb{Q}}(\beta) + o\left(\frac{1}{\ln\beta}\right)$ and $\psi\left(\frac{\beta}{2}\right) + \ln 2 = \ln(\beta) + o\left(\frac{1}{\ln\beta}\right)$. Further, substituting $2\mathrm{lgh}_{\mathbb{Q}}(\beta) = \ln\beta - \ln(2\pi e) + \frac{\ln\beta}{\beta} + o\left(\frac{\ln\beta}{\beta}\right)$, Equation (8) tells us we have a first-order relation

$$r + m = \beta + f^{(1)}_{r+m}(\beta)$$

where $f^{(1)}_{r+m}(\beta) = o(\beta)$. Substituting back into Equation (8) and simplifying we have the second-order relation

$$\begin{aligned} 2\ln\sigma &= -\ln(2\pi e) - \frac{f^{(1)}_{r+m}(\beta)+d}{\beta-d}(\ln\beta + o(\ln\beta)) + \frac{1}{d}\ln|\Delta_K| + \frac{2\alpha\ln q}{1+\alpha} + o(1) \\ \implies f^{(1)}_{r+m}(\beta) &= \ln\left(\frac{q^{\frac{2\alpha}{1+\alpha}}|\Delta_K|^{1/d}}{2\pi e\sigma^2}\right)\frac{\beta}{\ln\beta} + f^{(2)}_{r+m}(\beta) \end{aligned}$$

where $f^{(2)}_{r+m}(\beta) = o(\frac{\beta}{\ln\beta})$. We now define $C_{m+r} = \ln\left(\frac{q^{\frac{2\alpha}{1+\alpha}}|\Delta_K|^{1/d}}{2\pi e\sigma^2}\right)$ and $C_K = \frac{1}{d}\ln\frac{d^d}{|\Delta_K|} + \ln(2\pi e)$. Returning to Equation (8) once more,

$$\begin{aligned} \frac{C_K C_{m+r}}{\ln\beta} - \frac{f^{(2)}_{r+m}(\beta)}{\beta}(\ln\beta + o(\ln\beta)) &= o\left(\frac{1}{\ln\beta}\right) \\ f^{(2)}_{r+m}(\beta) &= C_K C_{m+r}\frac{\beta}{(\ln\beta)^2} + o\left(\frac{\beta}{(\ln\beta)^2}\right). \end{aligned}$$

Thus:

$$r+m = \beta + C_{r+m}\cdot\frac{\beta}{\ln\beta} + C_K C_{m+r}\cdot\frac{\beta}{(\ln\beta)^2} + o\left(\frac{\beta}{(\ln\beta)^2}\right). \tag{9}$$

C.2 Proof of Theorem 1

Removing asymptotically vanishing terms in the Module-GSA success condition gives

$$\ln(\sigma\sqrt{\beta}) = \mathrm{lgh}_{\mathbb{Q}}(\beta) - \frac{r+m-\beta}{\beta}\left(\mathrm{lgh}_{\mathbb{Q}}(\beta) + \frac{1}{2d}\ln\frac{|\Delta_K|}{d^d}\right) + \frac{\ln V_K}{r+m} + o\left(\frac{1}{\ln\beta}\right).$$

On the other hand, the GSA success condition for β_{eq} rearranges to

$$\ln(\sigma\sqrt{2}) + \frac{1}{2}\psi\left(\frac{\beta_{\mathrm{eq}}}{2}\right) < \mathrm{lgh}_{\mathbb{Q}}(\beta_{\mathrm{eq}}) - \frac{r+m+1-\beta_{\mathrm{eq}}}{\beta_{\mathrm{eq}}-1}\mathrm{lgh}_{\mathbb{Q}}(\beta_{\mathrm{eq}}) + \frac{\ln V}{r+m+1}$$

where $\ln V = \frac{r+m}{2d}\ln|\Delta_K| + m\ln q + \ln\sigma = \ln V_K - \frac{1}{2}|\Delta_K| - (d-1)\ln\sigma$. Simplifying lower order terms when β_{eq} grows[19],

$$\ln(\sigma\sqrt{\beta_{\mathrm{eq}}}) = \mathrm{lgh}_{\mathbb{Q}}(\beta_{\mathrm{eq}}) - \frac{r+m-\beta_{\mathrm{eq}}}{\beta_{\mathrm{eq}}}\mathrm{lgh}_{\mathbb{Q}}(\beta_{\mathrm{eq}}) + \frac{\ln V_K}{r+m} + o\left(\frac{1}{\ln\beta_{\mathrm{eq}}}\right)$$

Combining the relations satsified by β and β_{eq}, and using $2\mathrm{lgh}_{\mathbb{Q}}(x) = \ln x - \ln(2\pi e) + o\left(\frac{1}{\ln x}\right)$:

$$\frac{r+m-\beta}{\beta}(\ln\beta - C_K) = \frac{r+m-\beta_{\mathrm{eq}}}{\beta_{\mathrm{eq}}}(\ln\beta_{\mathrm{eq}} - C_{\mathbb{Q}}) + o\left(\frac{1}{\ln\beta}\right)$$

where $C_K = \frac{1}{d}\ln\frac{d^d}{|\Delta_K|} + \ln(2\pi e)$, and $C_{\mathbb{Q}} = \ln(2\pi e)$. Substituting the asymptotic expression for $r+m$ from Lemma 2 gives

$$\begin{aligned} C_{r+m} = \frac{\beta-\beta_{\mathrm{eq}}}{\beta_{\mathrm{eq}}}\ln\beta_{\mathrm{eq}} + C_{r+m}\frac{\beta\ln\beta_{\mathrm{eq}}}{\beta_{\mathrm{eq}}\ln\beta} - C_{\mathbb{Q}}C_{r+m}\frac{\beta}{\beta_{\mathrm{eq}}\ln\beta} \\ + C_K C_{r+m}\frac{\beta\ln\beta_{\mathrm{eq}}}{\beta_{\mathrm{eq}}(\ln\beta)^2} + o\left(\frac{1}{\ln\beta}\right) \end{aligned} \tag{10}$$

[19] As in the proof of Lemma 2, dominant term asymptotics will show that $r+m = \beta_{\mathrm{eq}} + o(\beta_{\mathrm{eq}})$.

Identification of constant terms gives:

$$\begin{aligned}\frac{\beta-\beta_{\text{eq}}}{\beta_{\text{eq}}}\ln\beta_{\text{eq}}+\frac{\beta\ln\beta_{\text{eq}}}{\beta_{\text{eq}}\ln\beta}-1&=o(1)\\(\beta-\beta_{\text{eq}})\ln\beta\ln\beta_{\text{eq}}+\beta\ln\beta_{\text{eq}}-\beta_{\text{eq}}\ln\beta&=o(\beta\ln\beta)\\(\beta-\beta_{\text{eq}})(\ln\beta_{\text{eq}}-1)&=o(\beta)\\\beta_{\text{eq}}&=\beta+o\left(\frac{\beta}{\ln\beta}\right)\end{aligned}$$

Hence:

$$\frac{\beta\ln\beta_{\text{eq}}}{\beta_{\text{eq}}\ln\beta}=\frac{\ln\beta_{\text{eq}}}{\ln\beta}+o\left(\frac{1}{\ln\beta}\right)$$

Replacing this expression in Equation 10, and simplifying negligible terms:

$$\begin{aligned}C_{r+m}\left(1-\frac{\ln\beta_{\text{eq}}}{\ln\beta}\right)&=\frac{\beta-\beta_{\text{eq}}}{\beta_{\text{eq}}}\ln\beta_{\text{eq}}+C_{r+m}\frac{C_K-C_{\mathbb{Q}}}{\ln\beta}+o\left(\frac{1}{\ln\beta}\right)\\C_{r+m}\ln\frac{\beta_{\text{eq}}}{\beta}&=\frac{\beta-\beta_{\text{eq}}}{\beta_{\text{eq}}}\ln\beta\ln\beta_{\text{eq}}+C_{r+m}(C_K-C_{\mathbb{Q}})+o(1)\end{aligned}$$

As $\beta_{\text{eq}}\sim\beta$, $\lim_{\beta\to+\infty}\ln\frac{\beta_{\text{eq}}}{\beta}=0$, and we complete the proof:

$$\beta_{\text{eq}}=\beta+(C_K-C_{\mathbb{Q}})\;C_{r+m}\;\frac{\beta}{(\ln\beta)^2}+o\left(\frac{\beta}{(\ln\beta)^2}\right)$$

D Cost comparison for Progressive BKZ/mBKZ

The time complexities of BKZ and Module-BKZ are dominated by calls to the SVP oracle, and so for fair comparison in §5.4 we aim to balance the total cost of oracle calls used by the structured and standard primal attacks.

It would be desirable to keep our comparison independent of the specific instantiation of the oracle. For primal attacks using a fixed-blocksize strategy this is possible: we compare equal blocksizes to balance the dimensions of lattices given to the oracle, and give Module-BKZ factor d more tours to balance the number of oracle calls.[20] However, for primal attacks using progressive blocksizes, we must understand how the cost of the oracle scales with dimension, necessitating assumptions on its instantiation.

We will assume solving SVP on a β-dimensional lattice has cost approximately $2^{c\beta}$ for some constant c, as is true for sieving algorithms. The literature typically estimates $c\approx 0.292$ [BDGL16] for classical sieving, with a quantum speedup $c\approx 0.265$ [LMvdP13, AGPS20].

Furthermore, we will assume solving SVP in a module lattice with $\mathbb{Q}$-dimension β has cost $2^{c_K\beta}$, where c_K is a constant depending on the underlying number field K. By ignoring the module structure, then certainly $c_K\leq c$, and it is possible this inequality is strict if the algebraic structure can be exploited to find shorter vectors faster in module lattices.

Claim. The cost of Progressive-BKZ[21] with $\beta=2,3,\ldots,B$ and τ tours at each blocksize, is approximately

$$C_{\text{BKZ}}=\tau\cdot N\cdot\frac{2^c}{2^c-1}\cdot 2^{cB},$$

where N is the rank of the lattice.

[20] Note that we assume that we are using both algorithms on lattices of the same $\mathbb{Q}$-dimension. In reality, the structured primal attack yields a lattice of dimension $(d-1)$ greater than the primal attack, but for the large ranks we consider, this difference can be ignored.

[21] Progressive BKZ is generally described starting from blocksize 3, but on an LLL-reduced basis, which means the initial step with blocksize 2 is implicit.

Proof. We sum a geometric progression. Each tour makes $N-\beta+1$ oracle calls with dimension β,[22] and so

$$\begin{aligned} C_{\mathrm{BKZ}} &= \tau \sum_{\beta=2}^{B} (N-\beta+1)2^{c\beta} \\ &= \tau \cdot \frac{2^{2c}}{2^c-1}\left(N(2^{c(B-1)}-1) - \frac{1-B2^{c(B-1)}+(B-1)2^{cB}}{2^c-1}\right) \\ &\approx \tau \cdot \frac{2^{2c}}{2^c-1} \cdot N\left(2^{c(B-1)}-1\right) \approx \tau \cdot N \cdot \frac{2^c}{2^c-1} \cdot 2^{cB} \end{aligned}$$

where we used the identity $\sum_{\beta=1}^{B-1} \beta\ 2^{c(\beta+1)} = \frac{2^{2c}(1-B\ 2^{c(B-1)}+(B-1)2^{cB})}{(2^c-1)^2}$ □

Claim. The cost of Progressive-mBKZ$_K$ with $\mathbb{Q}$-blocksizes $\beta = d, 2d, \ldots, B$ (where B is a multiple of d) and τ_K tours at each blocksize, is approximately

$$C_{\mathrm{mBKZ}} = \tau_K \cdot \frac{N}{d} \cdot \frac{2^{c_K d}}{2^{c_K d}-1} \cdot 2^{c_K B}.$$

where N is the $\mathbb{Q}$-dimension of the module lattice.

Proof. Each tour makes $\lfloor N/d \rfloor - d\beta + 1$ oracle calls with $\mathbb{Q}$-dimension β. Evaluating the resulting geometric progression is similar. □

To balance the costs $C_{\mathrm{mBKZ}} = C_{\mathrm{BKZ}}$, we should in theory set τ_K to be

$$\tau_K^* = d \cdot \tau \cdot \frac{2^c}{2^c-1} \cdot \frac{2^{c_K d}-1}{2^{c_K d}} \cdot 2^{(c-c_K)B}$$

but, to avoid providing concrete values for c and c_K, we simply use that $c_K \leq c$, corresponding to $\tau_K^* \geq d\tau$. In our experiments we take $\tau_K = d\tau$, which guarantees that $C_{\mathrm{mBKZ}} \leq C_{\mathrm{BKZ}}$.

E Choice of Embedding Submatrix

Similarly to the $\ell = 1$ case (the usual Bai–Galbraith embedding), viewing the lattice as an instance of f-unusual-SVP, we want to maximise f by maximising its $\mathbf{T}_\ell$-dependent factor

$$\frac{\lambda_1(\Lambda)}{|\det(\mathbf{T}_\ell)|^{\frac{1}{d(r_K+m_K)+\ell}}},$$

We note that $\lambda_1(\Lambda)$ and $\det(\mathbf{T}_\ell)$ are preserved when multiplying $\mathbf{M}_\ell$ left by matrices of the form

$$\mathbf{\Omega}_\ell = \begin{bmatrix} \mathbf{I}_{c(r_K+m_K)} & \mathbf{0} \\ \mathbf{0} & \mathbf{O}_\ell \end{bmatrix} \in \mathbb{R}^{(cr+cm+c\ell)\times(cr+cm+c\ell)}$$

where $\mathbf{O}_\ell \in \mathbb{R}^{\ell\times\ell}$ is an orthogonal matrix.[23] We may thus assume without loss of generality, using a singular value decomposition, $\mathbf{T}_\ell = \mathbf{DU}$ for $\mathbf{D}$ a diagonal matrix with coefficients $t_1, \ldots, t_\ell$, and $\mathbf{U}$ an orthogonal matrix. The expression we want to maximise may then be written as:

$$\frac{\prod_{k=1}^{\ell} |t_i|^{\frac{1}{d(r_K+m_K)+\ell}}}{\lambda_1(\Lambda)}$$

[22] We ignore the smaller-dimension oracle calls performed in the terminal blocks.

[23] $\mathbf{\Omega}_\ell \mathbf{M}_\ell = \begin{pmatrix} \mathbf{B} & \mathbf{b}_1 & \cdots & \mathbf{b}_\ell \\ \mathbf{0} & & \mathbf{O}_\ell\mathbf{T}_\ell & \end{pmatrix}$ and $\det(\mathbf{O}_\ell\mathbf{T}_\ell) = \det(\mathbf{T}_\ell)$. Moreover, $\mathbf{\Omega}_\ell^T\mathbf{\Omega}_\ell = \mathbf{\Omega}_\ell\mathbf{\Omega}_\ell^T = \mathbf{I}_{c(r_K+m_K)+\ell}$ so $\mathbf{\Omega}_\ell$ is orthogonal, and for any $\mathbf{x} \in \mathbb{R}^{c(r_K+m_K)+\ell}$, $\|\mathbf{\Omega}_\ell\mathbf{x}\| = \|\mathbf{x}\|$.

Writing $\mathbf{y}$ for a shortest vector of Λ, there exists $\mathbf{x} \in \mathbb{Z}^{d(r_K+m_K)+\ell}$ such that $\mathbf{y} = \mathbf{M}_\ell \mathbf{x}$. Writing $a = \|(\mathbf{B}, \mathbf{b}_1, \ldots, \mathbf{b}_\ell)\mathbf{x}\|^2$ and $x_1, \ldots, x_\ell$ for the last coordinates of $\mathbf{x}$, $\lambda_1(\Lambda)^2 = a + t_1^2 x_1^2 + \ldots + t_\ell^2 x_\ell^2$.[24] So, we wish to maximise:

$$\frac{\prod_{i=1}^{\ell} |t_i|^{\frac{1}{d(r_K+m_K)+\ell}}}{\sqrt{a + t_1^2 x_1^2 + \ldots + t_\ell^2 x_\ell^2}}$$

without knowing the distribution of a and $\mathbf{x}$. We pick $t_1 = \ldots = t_\ell = t > 0$, and as the above term is independent of the orthogonal matrix $\mathbf{U}$, we simply choose $\mathbf{U} = \mathbf{I}_\ell$. The maximum for

$$\frac{|t|^{\frac{\ell}{d(r_K+m_K)+\ell}}}{\sqrt{a + t^2(x_1^2 + \ldots + x_\ell^2)}}$$

will be reached when $t = \sqrt{\frac{a\ell}{d(r_K+m_K)b}}$, where $b = x_1^2 + \ldots + x_\ell^2$.

Moreover, we know that $\mathbf{v} = (\mathbf{e}, \mathbf{s}, t, 0, \ldots, 0)$ is a short vector of Λ as well as vectors of the form $(\mathbf{e}, \mathbf{s}, 0, \ldots, 0, t, 0, \ldots, 0)$, and possibly some of their linear combinations[25]. By reasoning analogous to our skewness analysis in §5, we expect $\lambda_1(\Lambda)$ to be close to $\|\mathbf{v}\|$, so we assume $\mathbf{v}$ is a shortest vector, in which case we have $b = 1$ and we expect $a = \sigma^2 d(r_K + m_K)$.

With these assumptions, the maximising choice of t is then $t = \sigma\sqrt{\ell}$.

[24] $\|\mathbf{T}_\ell \mathbf{x}\|^2 = \mathbf{x}^T \mathbf{U}^T \mathbf{D}^2 \mathbf{U} \mathbf{x} = \sum_{k=1}^{\ell} t_k^2 x_k^2$

[25] If they correspond to the action of one of the roots of unity of K on $\mathbf{v}$.