A refined analysis of the cost for solving LWE via uSVP

Shi Bai, Shaun Miller, Weiqiang Wen

A refined analysis of the cost for solving LWE via uSVP*

Shi Bai[1], Shaun Miller[1], and Weiqiang Wen[2]

[1] Department of Mathematical Sciences, Florida Atlantic University. United States.
shih.bai@gmail.com, shaunmiller2014@fau.edu
[2] Univ Rennes, CNRS, IRISA
weiqiang.wen@inria.fr

Abstract. The learning with errors (LWE) problem (STOC'05) introduced by Regev is one of the fundamental problems in lattice-based cryptography. One standard strategy to solve the LWE problem is to reduce it to a unique SVP (uSVP) problem via Kannan's embedding and then apply a lattice reduction to solve the uSVP problem. There are two methods for estimating the cost for solving LWE via this strategy: the first method considers the largeness of the gap in the uSVP problem (Gama-Nguyen, Eurocrypt'08) and the second method (Alkim et al., USENIX'16) considers the shortness of the projection of the shortest vector to the Gram-Schmidt vectors. These two estimates have been investigated by Albrecht et al. (Asiacrypt'16) who present a sound analysis and show that the lattice reduction experiments fit more consistently with the second estimate. They also observe that in some cases the lattice reduction even behaves better than the second estimate perhaps due to the second intersection of the projected vector with the Gram-Schmidt vectors. In this work, we revisit the work of Alkim et al. and Albrecht et al. We first report further experiments providing more comparisons and suggest that the second estimate leads to a more accurate prediction in practice. We also present empirical evidence confirming the assumptions used in the second estimate. Furthermore, we examine the gaps in uSVP derived from the embedded lattice and explain why it is preferable to use $\mu = 1$ for the embedded lattice. This shows there is a coherent relation between the second estimate and the gaps in uSVP. Finally, it has been conjectured by Albrecht et al. that the second intersection will not happen for large parameters. We will show that this is indeed the case: there is no second intersection as $\beta \to \infty$.

Keywords: Lattice-based cryptography · LWE · uSVP · Lattice reduction

* This work is in part supported through NATO SPS Project G5448 and through NIST awards 60NANB18D216 and 60NANB18D217, as well as the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701).

1 Introduction

A lattice is a discrete additive subgroup of $\mathbb{R}^n$. A lattice $\mathcal{L}$ of dimension n (of full-rank) can be described using a basis $\mathbf{B}$ consisting of linearly independent vectors $\mathbf{b}_1, \cdots, \mathbf{b}_n \in \mathbb{R}^n$ through integeral combinations $\mathcal{L}(\mathbf{B}) = \sum_{i=1}^{n} \mathbb{Z}\mathbf{b}_i$. Given a lattice basis $\mathbf{B}$ as input, one can apply lattice reduction algorithms such as [22,29,43,26,20,39] to find new bases made of relatively short and more orthogonal vectors. One quality measurement for a lattice basis $\mathbf{B}$ is the so-called Hermite factor $\mathrm{HF}(\mathbf{B}) = \|\mathbf{b}_1\|/(\mathrm{Vol}(\mathcal{L}(\mathbf{B})))^{1/n}$. Lattice reduction algorithms output reduced lattice bases with $\mathrm{HF}(\mathbf{B}) = \delta^n$ where δ is a function of the input parameter to the reduction algorithm. The number δ is also known as the root Hermite factor.

Lattices have attracted considerable interest in recent years as they can be used to construct cryptographic constructions (so-called lattice-based cryptography) which are believed to be quantum-resistant. Two fundamental computation problems in lattice-based cryptography are the short integer solution problem (SIS) [1,38] and the learning with errors problem (LWE) [40,41,37,17]. With parameters (m, n, q, B), the SIS problem is defined as follows: sample $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ (typically, $n \leq m$), the goal is to find non-zero $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{x} \equiv \mathbf{0} \pmod{q}$ and $\|\mathbf{x}\| \leq B$. Ajtai's seminal work [1] first established a worst-to-average connection for lattice-based primitives using the SIS problem. It then serves as a security foundation for numerous cryptographic primitives, including, among many others, hash functions [1] and signatures [25,35]. The LWE problem is introduced by Regev [40,41] and has been extensively used as a security foundation, for encryption schemes [41,25], fully homomorphic encryption schemes [18], signatures [25,21,35,10] and pseudo-random functions [15], and many others. The search version of the LWE problem with parameters (m, n, q, χ) is: sample $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ (typically $n \leq m$), the goal is to find the vector $\mathbf{s} \in \mathbb{Z}^n$ given samples $\mathbf{b}$ where $\mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q}$ and $\mathbf{e} \in \mathbb{Z}_q^m$ is a "short" error vector sampled from the given distribution χ. In this paper, we focus on χ which is a discrete Gaussian distribution of deviation αq. χ returns a vector $\mathbf{x} \in \mathbb{Z}_q^m$ with probability proportional to $\exp(-\|\mathbf{x}\|^2/(2\alpha^2 q^2))$.

Using lattice reduction, a standard method to solve the LWE problem is to first reduce it to an Unique Shortest Vector Problem (uSVP) via Kannan's embedding technique [30] and then apply a lattice reduction algorithm to solve the uSVP problem. For example, we describe the so-called primal lattice attack [2,7,5]. Given the matrix LWE instance $(\mathbf{A}, \mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q})$, we construct the lattice $\mathcal{L} = \{\mathbf{x} \in \mathbb{Z}^{m+n+1} \mid (\mathbf{A} \mid \mathbf{I}_m \mid \mathbf{b}) \cdot \mathbf{x} \equiv \mathbf{0} \bmod q\}$. This is a lattice of rank $d = m + n + 1$ and volume q^m. It is expected that $(\mathbf{s}, \mathbf{e}, -1)$ is the unique shortest vector in the lattice. Thus it boils down to find the shortest vector in the lattice which can be done by a lattice reduction algorithm. The goal is to estimate the cost of lattice reduction for solving the uSVP problem constructed from LWE.

There are two methods for estimating the cost for solving LWE using the aforementioned LWE-to-uSVP strategy. The first method is proposed by Gama

and Nguyen [23] and further investigated in subsequent works [2,6,28]. The main idea is to estimate the gap (between the first and second minima) in the uSVP lattice. As it is expected that $(\mathbf{s}, \mathbf{e}, -1)$ is the unique shortest vector in the lattice, the first minimum λ_1 of the uSVP lattice is about $\sqrt{\|\mathbf{e}\|^2 + \|\mathbf{s}\|^2}$. The second minimum λ_2 of the uSVP lattice is estimated from the Gaussian heuristic on random lattices: the expected first minimum of a lattice $\mathcal{L}$ of full rank d is about $\sqrt{d/(2\pi e)}\mathrm{Vol}(\mathcal{L})^{1/d}$. One assumes that the λ_2 of the uSVP lattice is about the same as the λ_1 of a random lattice with the same determinant and rank. Suppose a lattice reduction algorithm produces a reduced basis of root Hermite factor δ: for example, if a Block-Korkine-Zolotarev (BKZ) [42,44,43,26,20] algorithm of blocksize β is used, the root Hermite factor is about [19]:

$$\delta(\beta) \approx \left(\frac{\beta}{2\pi e} \cdot (\pi\beta)^{1/\beta}\right)^{\frac{1}{2(\beta-1)}}. \tag{1}$$

For large β, this is about $\beta^{1/(2\beta)}$ which we will use for asymptotic analysis. It then requires the uSVP gap $\gamma := \lambda_2/\lambda_1 \geq \tau \cdot \delta^d$ for a successful attack where τ is an experimental constant depending on the algorithm (and parameters). Finally, the running-time can be derived from the required δ given the gap γ which depends on the lattice reduction algorithm used. For the BKZ example, one can work out the blocksize β required and hence the running-time which is asymptotically $2^{O(\beta)}$ using the core-SVP model [7,3].

A second method is given in the New Hope key exchange paper [7]. Instead of looking at the gap of the uSVP directly, it considers the evolution of the Gram-Schmidt coefficients of the unique shortest vector in the BKZ tours. More precisely, it compares the expected length of the projection of the shortest vector orthogonally to the first $d - \beta$ Gram-Schmidt vectors with the length of $\mathbf{b}^*_{d-\beta+1}$ estimated using the GSA assumption. The justification is that, if this happens, the last β Gram-Schmidt coefficients of the shortest vector can be recovered during the local SVP of the last block.

These two estimates have been investigated extensively by Albrecht et al. in work [5]. They show that the lattice reduction experiments fits more consistently with the second estimate. They also present a sound analysis to show that, after the last β Gram-Schmidt coefficients of the shortest vector is recovered, a further size reduction is often sufficient to recover the complete secret. Interestingly, they also observe that in several cases the lattice reduction even behaves better than the second estimate for certain parameters. It is outlined that this may be caused by the occurrence of a second intersection of the projected vector with the Gram-Schmidt vectors.

1.1 Contribution

In this work, we revisit the analysis and experiments on estimating the cost for solving LWE via the uSVP approach. The experimental results are derived using the open-source lattice reduction libraries `FPLLL` and `FPYLLL` [46,47].

In Section 3, we first recall the two estimates from [23,7] and the analysis in [5]. Compared to [5], we expand the comparison of the two estimates with a larger set of LWE parameters (q, n, α). This complements the analysis and comparison in the Figure 1 of [5]. Furthermore, we verify the accuracy of the second estimate on the smaller dimension regime (Subsection 3.3), where the first estimate could lead to a smaller blocksize. For the second contribution (Subsection 3.4), we examine the projection length of the shortest vector on the reduced bases with different BKZ blocksize . This confirms that the assumption on the projection length is valid. Our third contribution (Section 4) is a concrete investigation of the uSVP gap in the embedded lattices with $\mu = 1$ and $\mu = \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$, given BDD instance $(\mathbf{B}, \mathbf{t})$ as input. It has been a common practice (e.g. [2,7]) to use $\mu = 1$ in the embedded lattice, albeit the reduction of BDD to uSVP [36] works only with $\mu = \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$ in theory. We show that the gap in the uSVP instances on average behaves much better than the worst-case guarantee. Finally, it has been observed in [5] that in several cases the lattice reduction even behaves better than the second estimate for some parameters. It is conjectured that the second intersection will not happen for large parameters. We show in Section 5 that this is true: we provide numerical experiments to confirm the impacts of the second intersection and present an analysis that the position/length of the second intersection approaches 0 as $\beta \to \infty$.

2 Preliminaries

In this section, we recall some basic facts on lattices, lattice reduction, and computational problems based on lattices. We first introduce the notations used throughout the paper.

Notations. We let lower-case bold letters denote column vectors and upper-case bold letters denote matrices. For a vector $\mathbf{x}$, we use $\|\mathbf{x}\|$ to denote its ℓ_2-norm. Similarly, a matrix $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$ is also presented in a column-wise way.

2.1 Euclidean lattices

Let $\mathbf{B} \in \mathbb{R}^{n \times n}$ be a full rank matrix. The lattice $\mathcal{L}$ generated by $\mathbf{B}$ is defined as $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$, and the matrix $\mathbf{B}$ is called a basis of $\mathcal{L}$ (or $\mathcal{L}(\mathbf{B})$). We let $\mathbf{B}^* = (\mathbf{b}_1^*, \cdots, \mathbf{b}_n^*)$ denote the Gram–Schmidt orthogonalization of $\mathbf{B}$. The determinant of a lattice $\mathcal{L}(\mathbf{B})$ is defined as $\mathrm{Vol}(\mathcal{L}(\mathbf{B})) = \prod_{i \le n} \|\mathbf{b}_i^*\|$. The ℓ_2-norm of a shortest non-zero vector in a lattice $\mathcal{L}$ is denoted by $\lambda_1(\mathcal{L})$ which is called the minimum of $\mathcal{L}$. This can be extended successively:

Definition 1 (Successive minima). *For any lattice $\mathcal{L}$, the i-th minimum $\lambda_i(\mathcal{L})$ is the radius of the smallest ball with center the origin and containing i linearly independent lattice vectors:*

$$\lambda_i(\mathcal{L}) = \inf\{r : \ \dim(\mathrm{span}(\mathcal{L} \cap \mathcal{B}(\mathbf{0}, r))) \ge i\}.$$

In subsequent sections, we will consider the ratio between λ_2 and λ_1. Minkowski's convex body theorem states that $\lambda_1(\mathcal{L}) \leq 2 \cdot v_n^{-1/n} \cdot \mathrm{Vol}(\mathcal{L})^{1/n}$ where v_n is the volume of an n-dimensional Euclidean ball of radius 1. The average version of the Minkowski's theorem is often known as the Gaussian heuristic: the λ_1 of a random n-dimensional lattice is asymptotically

$$\mathrm{GH}(\mathcal{L}) = v_n^{-1/n} \cdot \mathrm{Vol}(\mathcal{L})^{1/n}. \tag{2}$$

For $i \leq n$, we let $\pi_i(\mathbf{v})$ denote the orthogonal projection of $\mathbf{v}$ onto the linear subspace $(\mathbf{b}_1, \cdots, \mathbf{b}_{i-1})^{\perp}$. For $i < j \leq n$, we let $\mathbf{B}_{[i,j]}$ denote the local block $(\pi_i(\mathbf{b}_i), \cdots, \pi_i(\mathbf{b}_j))$, and $\mathcal{L}_{[i,j]}$ denote the lattice generated by $\mathbf{B}_{[i,j]}$.

2.2 Lattice problems

Two fundamental computation problems in lattice-based cryptography are the short integer solution problem (SIS) [1,38] and the learning with errors problem (LWE) [40,41,37,17]. They are defined as follows.

Definition 2 (Search $\mathrm{LWE}_{m,n,q,\chi}$). *With input parameters $n \geq 1$, modulus $q \geq 2$ and distribution χ, the search version of $\mathrm{LWE}_{m,n,q,\chi}$ problem consists of m samples of the form $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, with $\mathbf{a} \hookleftarrow U(\mathbb{Z}_q^n)$, $b = \langle \mathbf{a}, \mathbf{s} \rangle + e \pmod q$ and $e \hookleftarrow \chi$. Typically $m \geq n$. We say that an algorithm solves the search $\mathrm{LWE}_{n,q,\chi}^m$ if it outputs $\mathbf{s}$ with probability $\mathrm{poly}(1/(n \log q))$ in time $\mathrm{poly}(n \log q)$.*

If the number of samples is not restricted, we denote it as the $\mathrm{LWE}_{n,q,\chi}$ problem. In this work, χ is a discrete Gaussian of deviation αq. For convenience, we will also present the LWE in its the matrix form $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod q$.

A dual problem of LWE is the so-called short integer solution problem (SIS) [1,38]. We will mainly use its inhomogeneous version (ISIS) in this work.

Definition 3 (Search $\mathrm{ISIS}_{m,n,q,B}$). *Given $\mathbf{A}$ uniformly sampled from $\mathbb{Z}_q^{n \times m}$ and a vector $\boldsymbol{b} \in \mathbb{Z}^n$, find non-zero $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{x} \equiv \boldsymbol{b} \pmod q$ and $\|\mathbf{x}\| \leq B$. Typically $m \geq n$. If $\boldsymbol{b} = 0$, it is the $\mathrm{SIS}_{m,n,q,B}$ problem.*

Note that one can view the LWE problem $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ as an SIS-like problem by writing $\mathbf{A}' \cdot (\mathbf{s}|\mathbf{e}) \equiv \mathbf{0} \pmod q$ where $\mathbf{A}' = (\mathbf{A}|\mathbf{I})$. This also provide an alternative method for analyzing LWE via the SIS-like problem. The learning with errors problem (LWE) can be considered as an average version of the BDD problem:

Definition 4 (Bounded Distance Decoding: BDD_α). *Let $0 < \alpha < \frac{1}{2}$. Given a lattice basis $\mathbf{B}$ and a vector $\mathbf{t}$ such that $\mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq \alpha \cdot \lambda_1(\mathbf{B})$, find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ closest to $\mathbf{t}$. We will denote the α as the* gap *of the BDD_α problem.*

A dual problem of BDD is the so-called Unique Shortest Vector Problem (uSVP).

Definition 5 (Unique Shortest Vector Problem: uSVP$_\gamma$). *Let $\gamma \geq 1$. Given as input a lattice basis $\mathbf{B}$ such that $\lambda_2(\mathcal{L}(\mathbf{B})) \geq \gamma \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$, the goal is to find a non-zero vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ of norm $\lambda_1(\mathcal{L}(\mathbf{B}))$. We will denote the γ as the* gap *of the* uSVP$_\gamma$ *problem.*

In some cryptographic applications (e.g., lattice-based signatures [21,35,10]), it is preferred to use LWE problems where the secret $\mathbf{s}$ comes from the same distribution as the error $\mathbf{e}$. This is known as the *normal form LWE*. We will assume this is the case in this work. Notice that there exists a polynomial time reduction from LWE with secret from arbitrary distribution to LWE in normal form [8].

2.3 Lattice reduction

The security of lattice-based cryptography relies on the assumed hardness of solving the aforementioned geometric problems such as BDD and uSVP on high-dimensional lattices. The lattice reduction algorithms such as Block-Korkine-Zolotarev (BKZ) [42,44,43,20,27] are the most efficient methods for solving such problems currently known. Lattice reduction aims to compute a basis made of relatively short vectors from an arbitrary input basis. Quantitatively, one measure of quality is the so-called Hermite factor $\mathrm{HF}(\mathbf{B}) = \|\mathbf{b}_1\|/\mathrm{Vol}^{1/n}(\mathcal{L}(\mathbf{B}))$. Lattice reduction algorithms output reduced lattice bases with $\mathrm{HF}(\mathbf{B}) = \delta^n$ where δ is a function of the input parameter to the reduction algorithm. The δ is also known as the root Hermite factor (RHF).

We review some notions on lattice reduction. A lattice basis $\mathbf{B}$ is called size-reduced, if it satisfies $|\mu_{i,j}| \leq 1/2$ for $j < i \leq n$ where $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$. A basis $\mathbf{B}$ is HKZ-reduced if it is size-reduced and further satisfies:

$$\|\mathbf{b}_i^*\| = \lambda_1(\mathcal{L}_{[i,n]}),\ \forall i \leq n.$$

A basis $\mathbf{B}$ is BKZ-β reduced for blocksize $\beta \geq 2$ if it is size-reduced and satisfies:

$$\|\mathbf{b}_i^*\| = \lambda_1(\mathcal{L}_{[i,\min(i+\beta-1,n)]}),\ \forall i \leq n.$$

The work [19] shows that a BKZ-β reduced basis $\mathbf{B}$ satisfies $\|\mathbf{b}_1\| = \delta^n \mathrm{Vol}(\mathcal{L}(\mathbf{B}))$ where

$$\delta(\beta) \approx \left(\frac{\beta}{2\pi e} \cdot (\pi\beta)^{1/\beta} \right)^{\frac{1}{2(\beta-1)}}.$$

The Schnorr-Euchner BKZ algorithm [42,44,43] takes as inputs a blocksize β and a basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$ of a lattice $\mathcal{L}(\mathbf{B})$, and outputs a basis which is approximately BKZ-β-reduced, up to numerical inaccuracies. BKZ starts by LLL-reducing the input basis, then calls an SVP-solver of dimension β on consecutive local blocks $\mathbf{B}_{[k,\min(k+\beta-1,n)]}$ for $k = 1, \cdots, n-1$. This is referred to as one *BKZ tour*. Right after the local SVP at index k, if the found vector $\lambda_1(\mathcal{L}_{[k,\min(k+\beta-1,n)]}) < \|\mathbf{b}_k^*\|$, then BKZ updates the block $\mathbf{B}_{[k,\min(k+\beta-1,n)]}$ by

inserting the vector found between indices $k-1$ and k and does an LLL reduction. Otherwise, it moves to the next block. The procedure terminates when no change occurs at all during a tour. In practice, one prefers to terminate the BKZ when the changes between tours becomes less significant. This is called the early-abort BKZ [27]: Hanrot et al. showed that BKZ can be terminated long before its completion, while still providing bases of good quality.

It remains to estimate the running-time of BKZ-β given β. In the literature [6,20,7,3], there are several approaches to estimate the running-time of BKZ. The main differences come from two aspects: is sieving or enumeration used for the local SVP? and how many calls to the local SVP oracle are expected? For convenience, we will use the "core-sieving" model of [7,3]. Essentially it considers a single SVP call of dimension β using sieving, which can be modeled by a running-time of $2^{O(\beta)}$.

Heuristics. Lattice reduction algorithms and their analyses often rely on heuristic assumptions. A common heuristic is the aforementioned Gaussian heuristic (see Equation (2)). Let $\mathcal{L}$ be an n-dimensional lattice and $\mathcal{S}$ a measurable set in the real span of $\mathcal{L}$. The *Gaussian Heuristic* states that the number of lattice points in $\mathcal{S}$, denoted $|\mathcal{L} \cap \mathcal{S}|$, is about $\mathrm{vol}(\mathcal{S})/\mathrm{Vol}(\mathcal{L})$. In particular, taking $\mathcal{S}$ as a centered n-ball of radius R, the number of lattice points contained in the n-ball is about $V_n(R)/\mathrm{Vol}(\mathcal{L})$. Thus by setting $V_n(R) \approx \mathrm{Vol}(\mathcal{L})$, we see that $\lambda_1(\mathcal{L})$ is about $\mathrm{GH}(\mathcal{L}) = v_n^{-1/n} \cdot \mathrm{Vol}(\mathcal{L})^{1/n}$. Note that this is a factor of 2 smaller than the rigorous upper bound provided by Minkowski's theorem.

Another useful heuristic is the so-called *Geometric Series Assumption* (GSA) introduced in [45], which states that the Gram-Schmidt norms $\{\|\mathbf{b}_i^*\|\}_{i \leq n}$ of a BKZ-reduced basis behave as a geometric series, i.e., there is a constant $r > 1$ such that $\|\mathbf{b}_i^*\|/\|\mathbf{b}_{i+1}^*\| \approx r$ for all $i < n$.

2.4 Lattice attack for LWE

In this subsection, we recall several methods that are used to solve the LWE problem using lattices. In these methods, the main idea is to treat the LWE problem as a BDD/uSVP problem and then apply a lattice reduction algorithm to solve the BDD/uSVP problem.

The first method is to view the LWE problem as an ISIS-like problem: given $(\mathbf{A}, \mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod q)$ one can form an ISIS-like instance

$$(\mathbf{A}|\mathbf{I}_m) \begin{pmatrix} \mathbf{s} \\ \mathbf{e} \end{pmatrix} \equiv \mathbf{b} \pmod q$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix. We can then solve this ISIS instance using either a BDD solver or uSVP solver via embedding. For example, we may use the lattice generated by

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{A} & q\mathbf{I}_m \end{pmatrix}.$$

This is often known as the "primal attack". Usually matrix $\mathbf{A}$ has rank n. The $\mathcal{L}(\mathbf{B})$ is a lattice of rank $m+n$ and has volume q^m. We can then solve the BDD of $\mathcal{L}(\mathbf{B})$ with respect to the target point $\begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}$ which reveals $\begin{pmatrix} \mathbf{s} \\ -\mathbf{e} \end{pmatrix}$. Alternatively, we can reduce this BDD to uSVP; we will describe this method later.

The second method is to consider the lattice $\mathcal{L}_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} \equiv \mathbf{A}\mathbf{x} \pmod q, \forall \mathbf{x} \in \mathbb{Z}^n\}$. Note that the lattice $\mathcal{L}_q(\mathbf{A})$ contains a point which is close to the target point $\mathbf{b}$ within distance $\|\mathbf{e}\|$. One can hence solve the BDD of the lattice $\mathcal{L}_q(\mathbf{A})$ to the target point $\mathbf{b}$. The lattice $\mathcal{L}_q(\mathbf{A})$ has rank m and has volume q^{m-n}. This is equivalent to the "dual attack" where we multiply the left-kernel $\mathbf{A}^\perp$ of $\mathbf{A}$ on both sides of the equation $\mathbf{b} \equiv \mathbf{A}\mathbf{s} + \mathbf{e} \pmod q$. This leads to an ISIS-like problem of the form $\mathbf{A}^\perp\mathbf{b} \equiv \mathbf{A}^\perp\mathbf{e} \pmod q$ which we can solve using a BDD/uSVP solver.

These methods are sometimes equivalent, but not always, depending on the parameters given. For example, it has been investigated in [11,5] that for the binary secret LWE case, the first method leads to a better result since it uses the information about the smallness of $\mathbf{s}$. Furthermore, the allowed samples in cryptanalytic effort varies depending on the scheme considered. When there are not sufficiently enough samples, the first method might lead to a better complexity since it provides more "dimensions" for the lattice.

Reducing BDD **to u**SVP

We can solve the BDD using Kannan's embedding technique [30], Babai's nearest plane algorithm [9], or Lindner-Peikert's randomized nearest plane algorithm [33]. These algorithms have been further investigated by Liu and Nguyen [34] who show they can be considered as cases of pruned enumeration algorithms.

For the analysis of this paper we use Kannan's embedding technique. We describe it as follows. Given a BDD instance $(\mathbf{B}, \mathbf{t})$ where $\mathcal{L}(\mathbf{B})$ has rank d and $\mathbf{e}$ is the "shift", we consider the following basis matrix

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}.$$

This is a lattice of rank $d+1$ and volume $\mathrm{Vol}(\mathcal{L}(\mathbf{B}))$. Observe that

$$\mathbf{B}' \begin{pmatrix} \mathbf{x} \\ -1 \end{pmatrix} = \begin{pmatrix} \mathbf{B}\mathbf{x} - \mathbf{t} \\ -1 \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ -1 \end{pmatrix}.$$

Hence, the lattice generated by the columns of $\mathbf{B}'$ contains a short vector related to the potential solution of the BDD problem. Usually the lattice $\mathcal{L}(\mathbf{B}')$ derived from embedding is a uSVP problem of sufficiently large gap, albeit there is no theoretical proof for this. To solve this problem, we can use the aforementioned lattice reduction algorithms such as the BKZ algorithm.

In [36], Lyubashevsky and Micciancio provide a reduction, which can reduce any $\mathrm{BDD}_{1/\gamma}$ instance $(\mathbf{B}, \mathbf{t})$ to an $\mathrm{uSVP}_{\gamma/2}$ instance with basis:

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{t} \\ \mathbf{0} & \mu \end{pmatrix} \in \mathbb{Q}^{n+1},$$

with μ set to be the distance $d = \mathrm{dist}(\mathbf{t}, \mathcal{L}) \leq \lambda_1(\mathcal{L})/(2\gamma)$, where $\mathcal{L}$ is the lattice spanned by $\mathbf{B}$. In more detail, if $\mathbf{c}$ denotes a closest vector to $\mathbf{t}$ in $\mathcal{L}$ then it is shown that the vector $\mathbf{s}' = ((\mathbf{c} - \mathbf{t})^{\mathrm{T}}, -d)^{\mathrm{T}}$ is a shortest non-zero vector of lattice $\mathcal{L}'$ of basis $\mathbf{B}'$.

Later, Bai et al. [13] propose to preprocess the lattice $\mathcal{L}(\mathbf{B})$ using Khot's sparsification technique [31] before resorting to the Kannan's embedding: the component μ is decreased to be $\mathcal{O}(d/n)$, and the losing factor in the reduction is improved from 2 to $\sqrt{2}$.

However, on the practical side [2,7,5], one usually sets $\mu = 1$ in the embedded lattice and assumes there is no losing factor in the reduction. To be more precise, one assume that the first minimum and the second minimum of the embedded lattice are $\approx d$ and $\lambda_1(\mathcal{L}(\mathbf{B}))$, respectively. We assume this is true for the moment, but will have a detailed investigation on this topic in subsequent sections.

Other attacks

In this work, we focus on the expected cost of solving LWE by regarding it as BDD and then reducing it to uSVP. There are other types of algorithms for solving LWE such as the combinatorial attacks. These algorithms usually require exponential memory and a large number of LWE samples. We do not consider these attacks in this work but refer the reader to [16,4,32,12].

3 Revisiting the cost of solving uSVP

In this section, we first revisit the two approaches of [23,7] for estimating the cost of solving uSVP and the analysis in [5]. Then we expand the comparison in [5] of the two estimates with a larger set of LWE parameters. Furthermore, we verify the accuracy of the second estimate on the smaller dimension regime, where the first estimate could lead to a smaller blocksize.

3.1 Two estimates

Recall that we can view the LWE problem as a BDD problem. For simplicity, we will use the lattice $\mathcal{L}_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} \equiv \mathbf{A}\mathbf{x} \pmod q, \forall \mathbf{x} \in \mathbb{Z}^n\}$ defined in Subsection 2.4. The lattice $\mathcal{L}_q(\mathbf{A})$ with the target point $\mathbf{b}$ defines a BDD instance: note this is a $\mathrm{BDD}_{1/\gamma}$ instance with $\gamma = \lambda_1(\mathcal{L}_q(\mathbf{A}))/\|\mathbf{e}\|$. The lattice $\mathcal{L}_q(\mathbf{A})$ has rank m and volume q^{m-n}. By Gaussian Heuristic, we

have $\lambda_1(\mathcal{L}_q(\mathbf{A})) \approx \sqrt{\frac{m}{2\pi e}}\, q^{(m-n)/m}$. On the other hand, the LWE error $\mathbf{e}$ has length about $\sqrt{m}\alpha q$. Thus we obtain a $\mathrm{BDD}_{1/\gamma}$ instance where

$$\gamma \approx \frac{\min\left(q, \sqrt{\frac{m}{2\pi e}}\, q^{(m-n)/m}\right)}{\sqrt{m}\alpha q}. \tag{3}$$

For convenience, we assume that q is not too small and hence $\gamma \approx q^{-n/m}/\alpha$.

We first recall the estimate for solving uSVP by Gama and Nguyen [23] (we will refer to it as the *first estimate* or the *2008 estimate*). First, one assumes that the above $\mathrm{BDD}_{1/\gamma}$ reduces to uSVP_γ, where $\gamma \approx q^{-n/m}/\alpha$. Then Gama and Nguyen [23] show that the shortest vector in the uSVP_γ problem can be recovered as soon as $\gamma \geq \tau \cdot \delta^m$ where δ is root Hermite factor of the algorithm used. Here $\tau < 1$ is an empirical constant determined by experiments: it has been investigated that τ lies in between 0.3 and 0.4 when using the BKZ algorithm [2,5]. For simplicity, we will omit the constant τ in the asymptotic analysis (but set it to be 0.3 in actual experiments). As noted in Equation (1), the $\delta(\beta)$ is a decreasing function of β and therefore we want to maximize δ. The optimal m is asymptotically $\frac{2n\log q}{\log(1/\alpha)}$ which leads to maximum $\delta \approx \alpha^{\log\alpha/(4n\log q)}$. The running time of BKZ-β is $2^{O(\beta)}$ using the core-SVP model. In terms of LWE parameters this is asymptotically

$$\exp\left(c_t \cdot \frac{n\log q}{\log^2\alpha} \cdot \log\left(\frac{n\log q}{\log^2\alpha}\right)\right) \tag{4}$$

for some constant c_t.

In the New Hope key exchange paper [7], another method for estimating the cost for solving LWE is given. We will refer to it as the *second estimate* or the *2016 estimate*. Instead of looking at the gap of the uSVP directly, it considers the evolution of the Gram-Schmidt coefficients of the unique shortest vector in the BKZ tours. More precisely, it compares the expected length of the projected (expected) shortest vector $\mathbf{v} = (\mathbf{e}, -1)$ with the Gram-Schmidt lengths estimated by the GSA assumption. The key observation is that partial information of shortest vector $\mathbf{v}$ will be recovered in the last block, when the orthogonal projection of $\mathbf{v}$ to the first $d-\beta$ Gram-Schmidt vectors is shorter than the expected $\mathbf{b}^*_{d-\beta+1}$ predicated by the GSA assumption. Thus the success condition for recovering $(\mathbf{e}, -1)$ can be formulated as follows.

$$\sqrt{\beta}\alpha q \leq \delta^{2\beta-m} q^{(m-n)/m} \tag{5}$$

where δ depends on β. Here we simply take the rank of the lattice to be $m \approx d$.

These two estimates have been investigated extensively by Albrecht et al. in work [5]. They show that the lattice reduction experiments largely follow the behaviour expected from the second estimate. Furthermore, they also present

a sound analysis to show that, after the last β Gram-Schmidt coefficients of the shortest vector is recovered, a further size reduction is often sufficient to recover the complete secret immediately. In fact, this can happen at indices smaller than the $d - \beta + 1$. As noted in [5], they observe an interesting phenomenon that in several cases the lattice reduction even behaves better than the second estimate for some parameters: the BKZ algorithm recovers a projection $\pi_i(\mathbf{v})$ at index following a distribution with a center smaller than $d - \beta + 1$. It is outlined in [5] that this may be caused by the occurrence of a second intersection of the projected vector with the Gram-Schmidt vectors.

3.2 Comparison of estimates with various (n, q, α)

In this subsection, we expand the comparison in [5] on the two estimates with a larger set of LWE parameters. Note that a numerical comparison of two estimates is already given in the work [5]. Here we expand the range of the LWE parameters to the single-exponential regime: observe that the comparison in the Figure 1 of [5] fixes q, α and increases n. This compares the two estimates for LWE parameters in the super-exponential regime because of the estimate in Equation (4). Here we assumed that the optimal m in the 2006 estimate is asymptotically the same as the 2008 estimate. Note that the 2008 estimate (e.g. Equation (5)) can be re-formulated as

$$\beta^{1/(2\beta)} \leq \left(\frac{q^{-n/m}}{\alpha}\right)^{1/m} \beta^{1/(2m)}.$$

This can be compared to the uSVP gap argument in the 2008 estimate [23] where we have $\beta^{1/(2\beta)} \leq (q^{-n/m}/\alpha)^{1/m}$ instead. We want to minimize the β in Equation (5). This is a constraint optimization problem which seems tedious. Instead we find the optimal m and β numerically. In setting the LWE parameters (n, q, α), we maintain the relation that

$$\log q / \log^2 \alpha \cdot \log(n \log q / (\log^2 \alpha)) \tag{6}$$

being a constant c. Note that this corresponds to the multiplier in front of n in the Equation (4). This roughly means the running-time for solving LWE is asymptotically single-exponential.

We describe the parameters we used in the comparison. We set $c = 0.25$ and 0.35 respectively. For each c, we take $q = n^2$ and $q = n^4$ (thus four sets of parameters). Such parameters simulate commonly used conservative parameters (e.g. q not too large). Then we compute the corresponding α. For each set of parameters (n, q, α), we find the optimal m that leads to the smallest β using the 2016 estimate and the 2008 estimate (we set the empirical constant $\tau = 0.3$) respectively. We denote the smallest blocksize required from the two estimates as β_{2008} and β_{2016}. For each set of parameters, we plot the blocksize β required as an (increasing) function of n; we also plot the normalised blocksize difference which records $(\beta_{2008} - \beta_{2016})/\beta_{2008}$: this roughly illustrates the

"improvement percentage". If this value is negative, we simply denote it by 0 but we will further consider these cases later. We plot the comparison on the four sets of parameters in Figures 1-8.

It can be observed that the impacts of (the difference of) the two methods increases with the decrement of q. Similarly, the difference of the two methods increases with the decrement of α. This also confirms the comparison of the two methods in [5] in the single-exponential region.

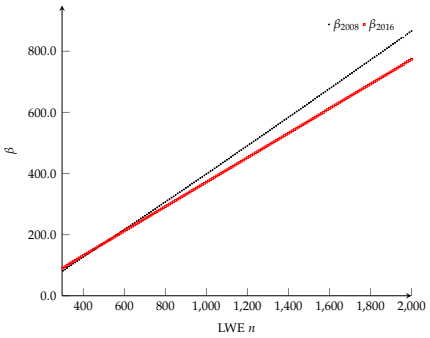

Fig. 1: Comparison of blocksize β of two estimates when $c = 0.25$ and $q = n^2$.

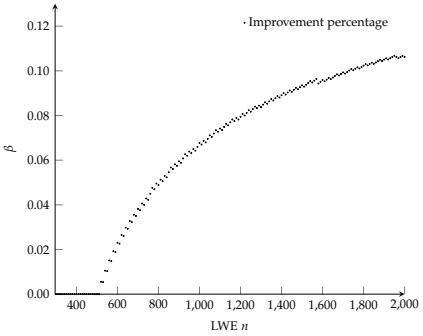

Fig. 2: Same as left hand side, but compares the improvement percentage of the blocksize.

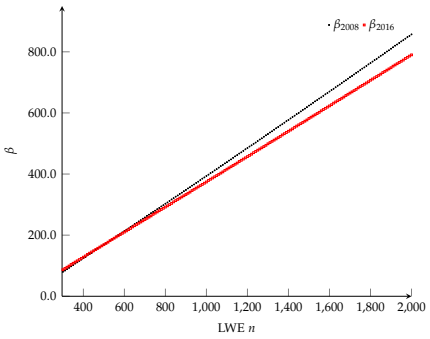

Fig. 3: Comparison of blocksize β of two estimates when $c = 0.25$ and $q = n^4$.

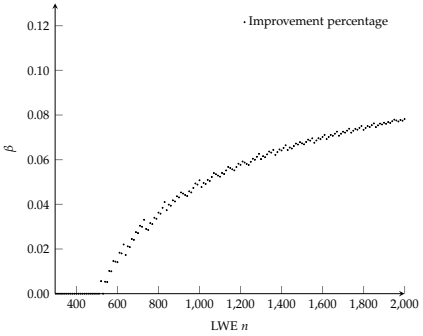

Fig. 4: Same as left hand side, but compares the improvement percentage of the blocksize.

3.3 Smaller dimension

Note that in the small dimension (in terms of LWE n) regime (some of which might be still relevant to practical schemes), the first estimate leads to a smaller

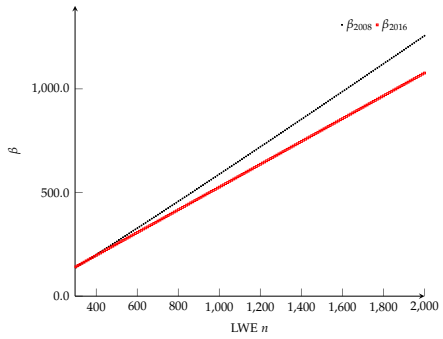

Fig. 5: Comparison of blocksize β of two estimates when $c = 0.35$ and $q = n^2$.

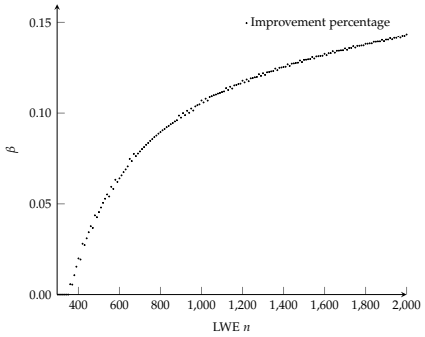

Fig. 6: Same as left hand side, but compares the improvement percentage of the blocksize.

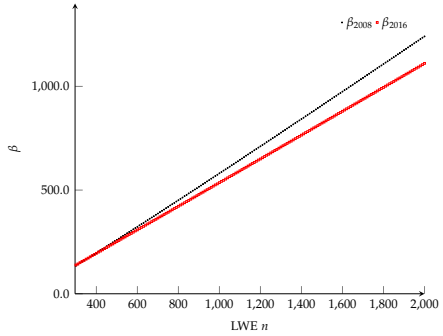

Fig. 7: Comparison of blocksize β of two estimates when $c = 0.35$ and $q = n^4$.

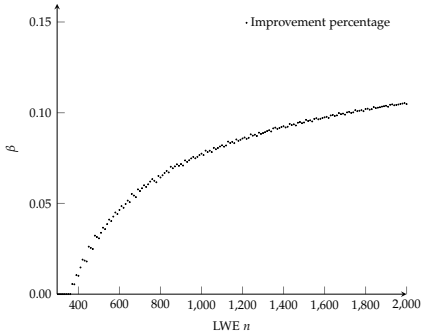

Fig. 8: Same as left hand side, but compares the improvement percentage of the blocksize.

blocksize. This is due to the empirical constant τ set to be 0.3. There might be a tendency to use the first estimate as it produces more conservative estimates. We further confirm the accuracy of the second estimate for these smaller dimensions. Note that for tiny blocksizes (e.g. $\beta \leq 30$), it has been observed in [14] that the Gaussian heuristic in local blocks is not accurate in BKZ; nor such blocksize matter the running-time of BKZ too much. Thus we do not consider these tiny blocksizes. We choose parameters n, q, α such that the blocksizes are ≥ 40 and compare the two methods in such region. Using the same approach as the last subsection, we set $c = 0.5$ and $q = n^2$. Then we find the corresponding α for the error rate. For each (n, q, α), we find the optimal m that leads to the smallest β using the 2016 estimate and the 2008 estimate respectively. For the 2008 estimate, we set the empirical constant $\tau = 0.3$: approximately we are comparing the two estimates in terms of $\delta^d \approx q^{-n/m}/(0.3\alpha)$ with $\delta^m \approx q^{-n/m}/\alpha\sqrt{\beta}$.

In Figure 9 we can observe, for small LWE dimension n, the first estimate gives a smaller blocksize due to the empirical constant 0.3. Then we look at the

concrete experiments with LWE parameters $n = 110$, $q = 12101$, $\sigma = \alpha q = 7.2$ of 100 instances. Using the 2008 estimate, the optimal $m = 277$ which leads to the $\beta = 39$. Using the 2016 estimate, the optimal $m = 294$ which leads to the $\beta = 66$. In Figure 10, the experiments using BKZ of various blocksize as well as different number of samples are tabulated. It can be seen that the 2016 estimate indeed provides a more accurate estimate: all BKZ instances using $\beta = 66$ succeed with $m = 294$ as predicated by the 2016 estimate. We note that many instances even succeeded with smaller blocksize $\beta = 60$. This is perhaps due to the second intersection phenomenon as observed in [5]. We will look at this phenomenon later.

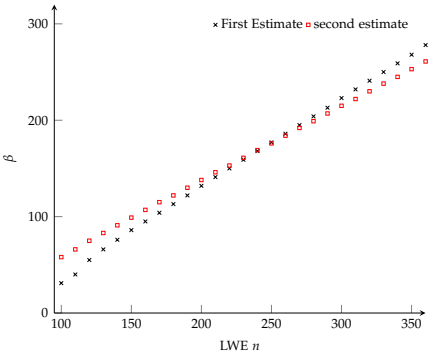

Fig. 9: Comparison of blocksize β of two estimates when $c = 0.5$ and $q = n^2$ for small n region.

LWE parameters: $n = 110$, $q = 12101$, $\sigma = 7.21$			
2008 estimate	Sample m	Blocksize β	Succ. prob.
(smallest $\beta = 39$	277	40	0%
with $m = 277$)	277	50	0%
2016 estimate	optimal m	blocksize β	Succ. prob.
(smallest $\beta = 66$	294	50	0%
with $m = 294$)	294	60	52%
	294	66	100%

Fig. 10: Experimental comparison of two estimate for small n region.

3.4 Further experiments on the projection length

The success condition for recovering the shortest vector in Equation (5) depends mainly on two heuristics: first, the norm of the Gram-Schmidt vectors in a BKZ reduced basis follows from the GSA assumption; second, the norm of the projection of the shortest vector onto the vector space spanned by the last β Gram-Schmidt vector is about $\alpha q\sqrt{\beta}$.

In practice, it is known [23,14] that the GSA assumption does not quite fit the BKZ experiments. However, the GSA assumption is optimistic from an attacker's point of view, which leads to a more conservative estimate. Hence we will assume this is the case. We will look at the second heuristic on the projection length. Denote the shortest vector to be $\mathbf{v}$. The heuristic on the project length essentially requires that $\mathbf{v}$, when expanded in terms of Gram-Schmidt vectors, have similar length on all components. This follows true if the Heuristic 2 described in work [24] is true: The distribution of the coordinates of the target vector $\mathbf{v}$, when written in the normalized Gram-Schmidt basis $(\mathbf{b}_1^*/\left\|\mathbf{b}_1^*\right\|, \mathbf{b}_2^*/\left\|\mathbf{b}_2^*\right\|, \cdots, \mathbf{b}_m^*/\left\|\mathbf{b}_m^*\right\|)$ of the input basis, looks like a uni-

formly distributed vector of norm $\|\mathbf{v}\|$. Observe that the heuristic depends on the shape of the input basis. For example, when the input basis is strongly reduced, the shortest vector $\mathbf{v}$ may already appear in the basis and hence the heuristic will not be true.

An experimental study has been presented in Figure 2 of [5] using 16 LLL reduced bases. We conduct further experiments on the length of projected shortest vector on BKZ reduced bases of various blocksizes. We use the same parameters as Figure 2 of [5]: we generate 200 LWE instances of $n = 65$, $m = 182$, $q = 521$ and $\sigma = 8/\sqrt{2\pi}$ (the results are averaged over these instances). We reduce the embedded bases using LLL and BKZ-β for $\beta = 10, 20, 30, 40, 45$. Note here we choose the largest blocksize to be 45 since this prevents the shortest vector from being recovered with high probability. Similarly, in the reduced bases, we do not consider those where the shortest vector has already been found. The experimental results are illustrated in Figures 11 and 12. It can be seen that the projection norms of the shortest vector indeed follow a similar shape in all LLL/BKZ-reduced bases. When the lattice is more reduced, the projected norm seems to follow more closely to the theoretical estimate except the last few indices. As a conclusion, it seems even plausible to use the theoretical estimate $\sqrt{m-i+1}\,\alpha q$ except for the last several indices. This might cause a problem for estimating the γ for the second intersection. We will consider such problem in a later section.

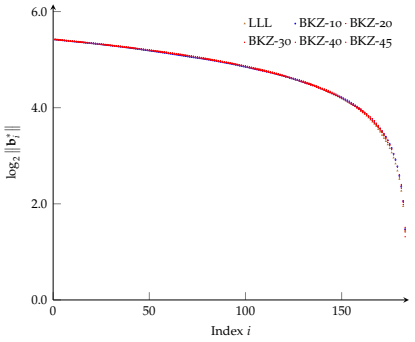

Fig. 11: Logarithmic norm of the projection of $\mathbf{v}$ on BKZ-β reduced bases for $\beta = 10, 20, 30, 40, 45$.

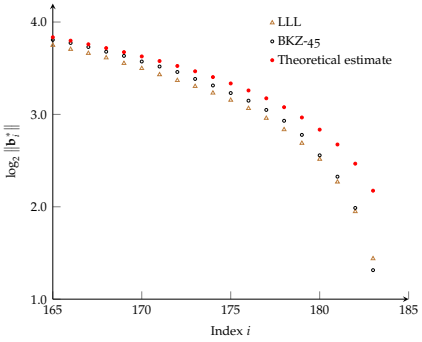

Fig. 12: Same as left hand side, but zoomed-in for only LLL and BKZ-45. Furthermore, theoretical estimate $\log_2(\sqrt{m-i+1}\,\alpha q)$ is plotted.

4 Gap in uSVP from LWE

In this section, we study the practical behavior of the reduction from the BDD problem to the uSVP problem. Note that in practice, we usually use the Kan-

nan's embedding with $\mu = 1$. However, in theory, it is not known that whether the gap γ of the embedded uSVP lattice in this case is optimal. But this seems to be the preferable setting in practice. As shown experimentally in [48], decreasing the embedding height is advantageous in solving LWE via the embedding technique. In this section we further explain why $\mu = 1$ is preferable by investigating the concrete gaps in the uSVP problem.

Let the BDD problem arise from LWE be $\mathrm{BDD}_{1/\gamma}$. We recall the reduction from $\mathrm{BDD}_{1/\gamma}$ to $\mathrm{uSVP}_{\gamma/2}$ by Lyubashevsky and Micciancio [36]. Given the $\mathrm{BDD}_{1/\gamma}$ instance $(\mathbf{B}, \mathbf{t})$, the following embedded lattice is contructed

$$\mathbf{B}' = \begin{pmatrix} \mathbf{B} & \mathbf{t} \\ \mathbf{0} & \mu \end{pmatrix} \in \mathbb{Q}^{n+1},$$

where μ is set to be the distance $d = \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$. Since this is a $\mathrm{BDD}_{1/\gamma}$ instance, we know that $d \leq \lambda_1(\mathcal{L})/\gamma$. Let $\mathbf{c} \in \mathcal{L}(\mathbf{B})$ denote a closest vector to the target point $\mathbf{t}$. Lyubashevsky and Micciancio [36] show that the vector $\mathbf{s}' = ((\mathbf{c} - \mathbf{t})^{\mathrm{T}}, -\mu)^{\mathrm{T}}$ is a shortest non-zero vector in the lattice $\mathcal{L}(\mathbf{B}')$ and other independent vectors are at least γ times larger than this. The reduction cares about the worst-case behaviors. In practice, it may be quite possible that all other independent vectors are more than $\gamma/2$ times larger and hence leads to a uSVP problem with larger gap. In fact, we will show that this is indeed the case in practice and investigate to what extent it is better than the $\gamma/2$-gap. Note that there is a natural upper-bound for the reduction. Precisely, the gap in the uSVP problem cannot be larger than $\sqrt{2}\gamma/2$ since a shortest vector in the BDD lattice also resides in the embedded lattice and $\mathbf{s}'$ has length $\sqrt{2}d/2$. On the other hand, in practice, we just take $\mu = 1$ in the embedded lattice. We assume that the vector $((\mathbf{c} - \mathbf{t})^{\mathrm{T}}, -1)^{\mathrm{T}}$ is a shortest non-zero vector in the lattice $\mathcal{L}(\mathbf{B}')$ and such that there is a sufficiently large gap between all other independent vectors and this shortest non-zero vector. In the 2008 estimate, this is equivalently assumed to be that the uSVP problem derived from $\mu = 1$ has a gap of γ (although this is not supported theoretically in the worst case). In fact, such γ-gap already implies the reduction has reached its natural upper bound – note that the shortest vector in the given BDD lattice $\mathcal{L}(\mathbf{B})$ is about γ times larger than d as defined.

In this section, we investigate concretely the gap in the uSVP problem in experiments. Perhaps surprisingly, we show that the gap in the uSVP instance are somewhat close to the upper-bound γ in practice, even though this is not guaranteed in the worst-case. This also explains that why it is preferable to use $\mu = 1$ in practice. We set up the following experiments to investigate the gap in the resulted uSVP instance in practice. For each set of parameters, we generate 100 LWE instances. For each instance, we construct the embedded lattices in two ways, with $\mu = 1$ and $\mu = d$ where $d = \lfloor \|\mathbf{e}\| \rceil$. In experiments, we compute and compare the gaps in the resulted uSVP instances.

We explain the notations in Table 1. For each parameter n, m, q in LWE, we use error deviation $\sigma = 3.1925 \approx \frac{8}{\sqrt{2\pi}}$. For each LWE/BDD instance,

n	m	q	BDD lattice		uSVP lattice $\mu = 1$			uSVP lattice $\mu = \|\mathbf{e}\|$		
			Theory	Experiment	Theoretical upper	Experiment	Ratio	Theoretical upper	Experiment	Ratio
16	32	1031	2.71	2.78	2.78	$\lesssim 2.55$	0.92	1.97	$\lesssim 1.96$	0.71
16	48	1031	8.40	8.49	8.49	$\lesssim 7.81$	0.92	6.00	$\lesssim 5.99$	0.71
32	48	8101	1.65	1.68	1.68	$\lesssim 1.58$	0.94	1.19	$\lesssim 1.19$	0.71
32	64	8101	7.23	7.33	7.33	$\lesssim 6.95$	0.94	5.18	$\lesssim 5.16$	0.70

Table 1: Experimental comparison on the gap of uSVP derived from two embeddings.

we calculate the theoretical gap in the BDD problem from $\min\left(q, (\Gamma(1+m/2)^{1/m})/\sqrt{\pi}\cdot q^{(m-n)/m}\right)/(\sigma\sqrt{m})$. Note that we can measure in a better way: since we know the errors, we use the average norm of the errors in the denominator (instead of the estimate $\sigma\sqrt{m}$). This is tabulated in the **"Theory"** sub-column under "BDD". Then we use BKZ_m to find the $\lambda_1(\mathcal{L}(\mathbf{B}))$ and divide that by the norm of error in LWE. This is recorded in the **"Experiment"** sub-column under "BDD". Note that the experimental values obtained is slightly larger than the theory; this is perhaps due to the solver only finding the approximate shortest vector in practice. Then we construct the embedded uSVP lattices with $\mu = 1$ and $\mu = d$, respectively. The sub-columns **"Theoretical upper"** under "uSVP lattice" denote the upper bound of the gap in the uSVP instances one can achieve using the values in the **"Experiment"** (not "Theory") sub-column under "BDD", for each type of embedding, respectively. For example, the experiment value 2.78 under $n = 16, m = 32, q = 1031$ implies that the corresponding uSVP instances with $\mu = \|\mathbf{e}\|$ can at most have a gap of 1.97. The sub-column **"Experiment"** under "uSVP lattice" gives the experimental values for the gaps between the norm of a second shortest vector and $\left\|(\mathbf{e}^{\mathrm{T}}, -\mu)^{\mathrm{T}}\right\|$. Note that here we approximate the norm of a second shortest vector by considering the second shortest vector in a reduced basis using BKZ of blocksize m. This is not necessarily the λ_2 but hopefully a close approximation. Thus we denote "$\lesssim$" in the table. For the lattice reduction, we use BKZ in `FPLLL` until exhaustion with full enumeration for $m = 32$ and pruned enumeration for other m. The sub-column **"Ratio"** under "uSVP lattice" computes the ratio between the uSVP gap and the BDD gap. That is, it reflects the practical behavior of the reduction from $\mathrm{BDD}_{1/x}$ to uSVP_y where the sub-column "Ratio" is computed as y/x. The larger the ratio, the better (larger gap) the uSVP instance is. All the figures in the table are averaged over 100 instances.

From a theoretical perspective, it is perhaps surprising to see that the BDD-uSVP reduction works pretty well in practice with both μ. In particular, with $\mu = 1$, it seems that $\mathrm{BDD}_{1/\gamma}$ already reduces to $\mathrm{uSVP}_{0.9\gamma}$ in practice. In theory for such case ($\mu = 1$), it is possible that there exists a lattice point $\mathbf{c}' \in \mathcal{L}(\mathbf{B})$ that is closer to $k\cdot\mathbf{t}$ for some multiple k, and therefore $(\mathbf{c}' - k\cdot\mathbf{t}, -k)$ decreases the desirable gap. However, experiments in Table 1 seems to imply that such bad points are rare in practice. Note that such cases can be provably eliminated by setting a larger $\mu = \|\mathbf{e}\|$ as shown in [36]. Specifically for such μ, it

is guaranteed that the uSVP gap is $\gamma/2$ (from $\text{BDD}_{1/\gamma}$) in the worst case. Similarly, the practical/average behavior seems to be much better: with $\mu = \|\mathbf{e}\|$, the $\text{BDD}_{1/\gamma}$ problem reduces to $\text{uSVP}_{0.7\gamma}$ in practice.

We do not know how to explain such average behavior in theory. It may be related to the difference on the natural upper-bounds in two embeddings: with $\mu = 1$, the natural upper-bound of the gap in the uSVP problem is γ. This is larger than that (e.g. $\gamma/\sqrt{2}$) derived from the lattice using $\mu = \|\mathbf{e}\|$. Thus it may be due to a larger upper-bound providing larger "room" for the reduction, together with annoying "extremely close" lattice points (to multiple of target vector $\mathbf{t}$) being rare in practice. It may be interesting to further investigate this, e.g. by trying more μ between 1 and $\|\mathbf{e}\|$ and observe the impacts to the uSVP gap. We leave more investigations on this for future work.

So far, we've only discussed the gap appeared in the embedded uSVP instance under different embedding parameters. We further look at the impacts on the cost estimate under different embedding heights. In the 2008 estimate, it is assumed that given as input a $\text{BDD}_{1/\gamma}$ problem, one can reduce to a uSVP_{γ} problem. Then the root Hermite factor δ can be derived from the gap γ and hence the blocksize & running-time. It is also natural to see that when using the 2008 estimate, it is preferable to use $\mu = 1$ since it leads to a larger gap in the uSVP problem. In the 2016 estimate, the gap of the uSVP problem is not used explicitly. But one can see that the estimate is asymptotically equivalent to $\delta^m \leq \sqrt{\beta}\frac{\sqrt{m}q^{(m-n)/m}}{\|(\mathbf{e}|\mu)\|}$. The fractional part of the equation corresponds to the gap in the uSVP problem. Note that the difference on the gap using $\mu = 1$ and $\mu = \|\mathbf{e}\|$ is at most a scaling factor of $\sqrt{2}$. It seems to be a small factor however it may affect the concrete security level of schemes with moderate size.

5 Second intersection

An interesting phenomenon observed in [5] shows that in several cases the lattice reduction behaves even better than the 2016 estimate for some parameters. First, the BKZ algorithm recovers a projection $\pi_i(\mathbf{v})$ at index following a distribution with a center below $d - \beta + 1$. After that, a size reduction usually immediately recovers the full secret. It is outlined in [5] that this may be caused by the occurrence of a second intersection of the projected vector with the Gram-Schmidt vectors. For example, to solve LWE parameter $n = 65, m = 182, q = 521$ and $\alpha q = 8/\sqrt{2\pi}$, it runs BKZ with blocksize $\beta = 56$ according to Equation (5). Since $\beta = 56$ satisfies Equation (5), a projection of our error should be found at index $d - \beta + 1 = 128$, recovering the last 56 coefficients of the error which leads to size reduction recovering the rest. In experiments the projection is found earlier (at index ≈ 124.76) and the coefficients of the error are found after one more call to size reduction. Second, the blocksize required to recover the secret (on average) is actually smaller than that estimated from Equation (5). For the LWE parameter mentioned above, it

requires to run BKZ using blocksize 56 according to Equation (5). However, as noted in [5], using blocksize 51 is sufficient to recover more than half of the instances. Some justification has been outlined in Subsection 4.3 of [5], mainly on the size reduction at index $\leq d - \beta + 1$. We will provide a refined analysis of why a smaller blocksize may work.

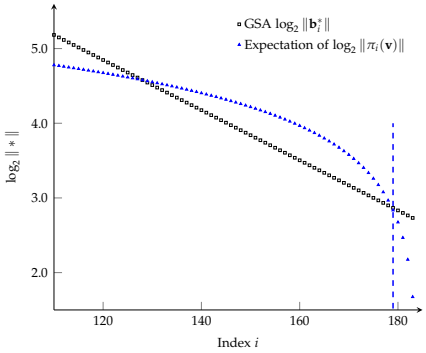

Fig. 13: Comparison between G-S norms of BKZ_{56} under GSA and the expected length of $\pi_i(\mathbf{v})$.

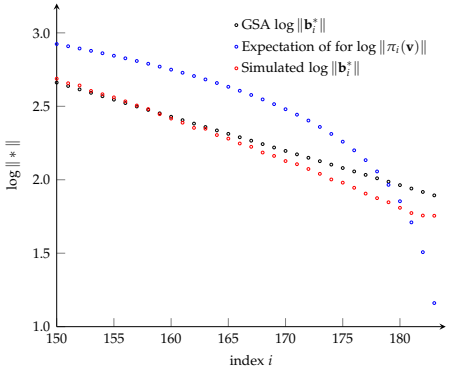

Fig. 14: Same as LHS, but zoomed-in to the last several indices. Furthermore, a BKZ simulator is used to estimate the $\log \|\mathbf{b}_i^*\|$.

We first recall the phenomenon in more detail as well as a brief explanation given in [5]. According to Equation (5), the projection of the shortest vector should be recovered at position $d - \beta + 1$ when running the BKZ with blocksize β on the uSVP instance over a d-dimensional lattice (recall that in our description, the $d = m + 1$). However, it is observed that the existence of a second intersection on the expected projection length of the shortest vector and the Gram-Schmidt norms under GSA assumption may speed-up the recovery of $\mathbf{v}$. For example, Figure 13 compares the (logarithmic) Gram-Schmidt norms of BKZ_{56} reduced basis under GSA assumption and the expected length of $\pi_i(\mathbf{v})$. Note there are 5 indexes in which $\|\pi_i(\mathbf{v})\|$ is smaller than the Gram-Schmidt norms, thus in this case, we denote $\kappa = 5$. In particular, after the second intersection, the expected length of $\pi_i(\mathbf{v})$ will be less than the $\|\mathbf{b}_i^*\|$ for κ indexes in the end. Hence the projection is likely to be the smallest vector of the projected lattice $\mathcal{L}(\pi_{d-\kappa+1}(\mathbf{b}_{d-\kappa+1}), \cdots, \pi_{d-\kappa+1}(\mathbf{b}_d))$ of dimension κ. The SVP oracle will find this projection and the BKZ algorithm will then insert it at index $d - \kappa + 1$. As a result, $\mathbf{b}_{d-\kappa+1}$ is updated to be (the lifted vector of) the projection of the vector $\mathbf{v}$ over the last κ Gram-Schmidt vectors. Further, it is likely that $\pi_{d-\beta-\kappa+1}(\mathbf{v})$ is the shortest vector of the projected lattice $\mathcal{L}(\pi_{d-\beta-\kappa+1}(\mathbf{b}_{d-\beta-\kappa+1}), \cdots, \pi_{d-\beta-\kappa+1}(\mathbf{b}_{d-\kappa+1}))$ of size β after which $\mathbf{v}$ can be recovered by a size reduction according to [5]. Therefore, assuming a projection of our vector $\pi_{d-\kappa+1}(\mathbf{v})$ has already been found, an SVP oracle will find $\pi_{d-\beta-\kappa+1}(\mathbf{v})$ in the lattice $\mathcal{L}(\pi_{d-\beta-\kappa+1}(\mathbf{b}_{d-\beta-\kappa+1}), \cdots, \pi_{d-\beta-\kappa+1}(\mathbf{b}_{d-\kappa+1}))$.

5.1 On smaller blocksize

A related interesting phenomenon is that often a smaller blocksize may be already sufficient to solve the uSVP problem. This has been observed in [5] where a blocksize of $\beta' = \beta - \kappa$ is sufficient to recover the secret with high probability. We give a heuristic justification of this based on the second intersection. Suppose now β is the smallest blocksize that satisfies Equation (5) with a nonzero κ depending on β.

Denote $\beta' = \beta - \kappa$. Suppose $\mathrm{BKZ}_{\beta'}$ is run (instead of BKZ_{β}). For convenience, let δ_β denote the value of δ given blocksize β. Let κ' be the amount of indices where the projection of $\mathbf{v}$ is smaller than the GSA predicated Gram-Schmidt norm. Due to the second intersection, a projection of $\mathbf{v}$ is likely to be found at index $d - \kappa' + 1$ so after SVP the vector $\mathbf{b}_{d-\kappa'+1}$ will contain the last κ' coefficients of $\mathbf{v}$. Therefore the norm of $\mathbf{v}$, if decomposed in terms of the Gram-Schmidt vectors $\mathbf{b}_i$, will concentrate on the first $d - \kappa' + 1$ components. More precisely, $\|\mathbf{v}\|^2 = \sum_{i=1}^{d-\kappa'+1} c_i^2 \left\|\mathbf{b}_i^*\right\|^2$ where c_i are the coefficients in the decomposition. Following the same reasoning as in [5], we look at the β' dimensional lattice $\mathcal{L}(\pi_{d-\beta'-\kappa'+1}(\mathbf{b}_{d-\beta'-\kappa'+1}), ..., \pi_{d-\beta'-\kappa'+1}(\mathbf{b}_{d-\kappa'+1}))$. If the projected shortest vector has a smaller norm than the GSA predicated norm of blocksize β', then we would be able to recover the last $\beta' + \kappa$ coefficients. The success condition can be phrased as

$$\sqrt{\beta' + \kappa'}\alpha q \leq \delta_{\beta'}^{2\beta'-d+2\kappa'} \mathrm{Vol}(\mathcal{L})^{1/d}. \tag{7}$$

Equation (7) is sometimes satisfied, but not always, depending on the relation between κ and κ'. It seems plausible to assume that $\kappa' \approx \kappa$ for the analysis, albeit this may not be true in practice. (This can be seen from experiments the newly found β' will not recover as many error vectors as the original β. For example, $\beta' = 51$ in the aforementioned LWE parameters can only recover half of the instances.) Note that if $\kappa \approx \kappa'$, the left-hand side of Equation (7) is the same as $\sqrt{\beta}\alpha q$ and the right-hand side is larger, hence $\pi_{d-\beta'-\kappa'+1}(\mathbf{v})$ is the shortest vector in the local lattice. By recovering $\beta' + \kappa$ coefficients of $\mathbf{v}$, a following size reduction will find the rest with a high probability.

5.2 Experiments on κ

In the experiments to follow, we consider the last projection of our vector $\mathbf{v}$ that was found before it is completely recovered in the next tour by size reduction. This confirms the existence of κ in practice. With LWE parameters $n = 65, m = 182, q = 521$ in both parameter sets, we consider two different choices of αq that produces different κ. The first is $\alpha q = 3.192$ and requires $\beta = 56$ while the second is $\alpha q = 2.469$ and requires $\beta = 42$. We run 800 instances in total and take the average for both parameter sets. The distribution of κ found are plotted in Figures 15 and 16. The y axis represents the counts over 800 where a projection of $\mathbf{v}$ was found at index $d - \kappa + 1$ before the tour it was completely recovered and the x axis is the value κ. In both cases, we did

not consider projections of $\mathbf{v}$ that were found at an index less than or equal to $d - \beta + 1$ as this will probably be where $\mathbf{v}$ is recovered by size reduction. The experiment that required $\beta = 56$ was allowed to run for at most 20 tours while the experiment requiring $\beta = 42$ is allowed 60 tours.

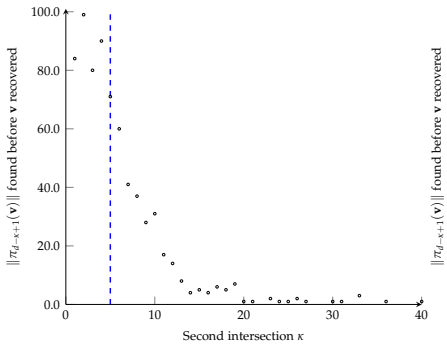

Fig. 15: Blocksize $\beta = 56$ required in Equation (5) and $\kappa = 5$.

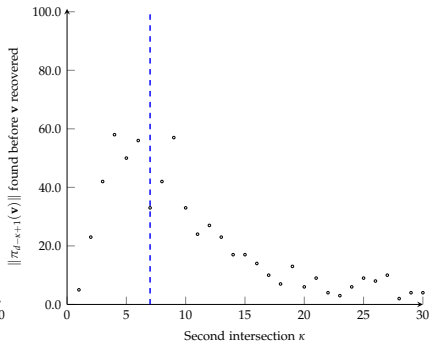

Fig. 16: Blocksize $\beta = 42$ required in Equation (5) and $\kappa = 7$.

We notice that the experimental values for κ indeed follow approximately from the theoretical predicate from Equation (5). However, we also notice that the experimental value for κ seems to be slightly less than the predicted value. This could be due to the inaccuracy of GSA when predicting the length of the last few projections. It is known that the simulator-based approach [23,14] provides a better estimation for the behavior of the lengths $\|\mathbf{b}_i^*\|$. We considered the average simulated $\|\mathbf{b}_i^*\|$ over 1000 instances with blocksize 56 and 200 tours. By comparing the simulator to the expected length of our projection (see Figure 14), we see that fewer projections of $\mathbf{v}$ are below the simulator after the second intersection: There are 3 (resp. 5) indexes in which $\|\pi_i(\mathbf{v})\|$ is smaller than the simulator's (resp. GSA's) value for $\|\mathbf{b}_i^*\|$ (comparing Figure 15 with Figure 14).

5.3 Convergence of κ

It has been conjectured [5] that the second intersection will not happen for cryptographic meaningful parameters. We first show that the position of the second intersection approaches 0 as $\beta \to \infty$. We will also provide a numerical analysis for the index of the second intersection using both GSA assumption and simulator. We first take the logarithm of both the Gram-Schmidt norm at index x and the norm of $\pi_x(\mathbf{v})$:

$$\log(\pi_x(\|\mathbf{v}\|)) \approx \log(\sqrt{d - x + 1} \cdot \alpha q),$$
$$\log(\|\mathbf{b}_x^*\|) \approx (x - 1)\log(\alpha) + \log(\|\mathbf{b}_1\|)$$

where $\alpha \approx \delta^{-2}$ is the constant ratio in GSA. Note that $\|\mathbf{b}_1\| \approx \delta^d \mathrm{Vol}(\mathcal{L})^{1/d}$. Assuming Equation (5) is satisfied so that $\alpha q \approx \delta^{2\beta-d}\mathrm{Vol}(\mathcal{L})^{1/d}/\beta^{1/2}$, the inequality can be represented as

$$\log(\frac{\kappa}{\beta}) \leq -4\log(\delta)(-\kappa+\beta) \tag{8}$$

where $\kappa = d - x + 1$. If there a nontrivial second intersection, the above relation has to be true for at least $\kappa = 1$. Using $\delta \approx v_\beta^{-1/(\beta(\beta-1))}$, one could see that for large enough blocksize, this relation can not be satisfied and hence the second intersection will not happen for large blocksize. This shows that the second intersection approaches 0 as $\beta \to \infty$. Further, we numerically investigate the evolution of κ in terms of β using relation (8). Figure 17 considers the values of κ given by relation (8) for different values of β. Notice that Figure 17 shows that $\beta = 278$ is the smallest blocksize where κ already becomes 0. This suggests there is no second intersection when $\beta \geq 278$ is needed to satisfy equation Equation (5). However, this could be an over-estimate from the attacker's point of view since the GSA assumption is used here.

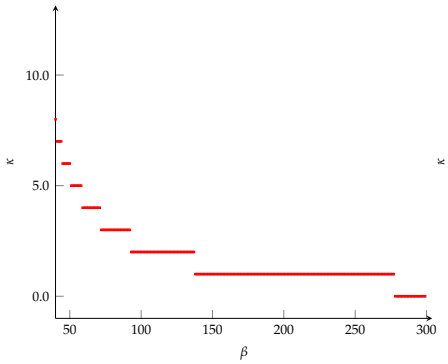

Fig. 17: Maximal κ satisfying Equation (8) given β.

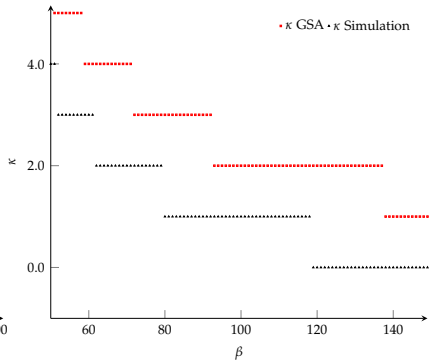

Fig. 18: Maximal κ satisfying Equation (8) given β.

To get a more accurate estimation of the value κ, we further compare that with the BKZ simulator. The next figure considered several different parameter sets ($n = 65, m = 182, q = 521$) only varying in αq and necessary β (averaged over α and LWE instances). We simulate 200 tours of BKZ-β using the BKZ simulator and averaged 1000 instances of each parameter set. Figure 18 shows that the value of κ derived by comparing the simulated $\|\mathbf{b}_i^*\|$ to $\|\pi_i(\mathbf{v})\|$ suggests there is no second intersection for blocksizes larger than 120. One can also see this produces slightly smaller κ for a given β than the comparison assuming GSA. This seems reasonable since the GSA assumption is known to be optimistic from an attacker's point of view. In conclusion, this further suggests that a second intersection will only affect the results of running BKZ-β on smaller parameter sets.

Acknowledgments

We thank the reviewers for their valuable comments and suggestions. The authors would like to acknowledge the use of the services provided by Research Computing at the Florida Atlantic University.

References

1. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108. ACM Press, May 1996.
2. M. Albrecht, R. Fitzpatrick, and F. Göpfert. On the efficacy of solving lwe by reduction to unique-svp. In *Proc. of ICICS*, volume 8233 of *LNCS*, pages 293–310. Springer, 2013.
3. M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wunderer. Estimate all the {LWE, NTRU} schemes! In D. Catalano and R. D. Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 351–367. Springer, 2018.
4. M. R. Albrecht, J.-C. Faugère, R. Fitzpatrick, and L. Perret. Lazy modulus switching for the BKW algorithm on LWE. In H. Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 429–445. Springer, Heidelberg, Mar. 2014.
5. M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In T. Takagi and T. Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 297–322. Springer, Heidelberg, Dec. 2017.
6. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *J. Mathematical Cryptology*, 9(3):169–203, 2015.
7. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In T. Holz and S. Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343. USENIX Association, Aug. 2016.
8. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Proc. of CRYPTO*, volume 5677 of *LNCS*, pages 595–618. Springer, 2009.
9. L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
10. S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. In J. Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *Lecture Notes in Computer Science*, pages 28–47. Springer, Heidelberg, Feb. 2014.
11. S. Bai and S. D. Galbraith. Lattice decoding attacks on binary LWE. In W. Susilo and Y. Mu, editors, *ACISP 14: 19th Australasian Conference on Information Security and Privacy*, volume 8544 of *Lecture Notes in Computer Science*, pages 322–337. Springer, Heidelberg, July 2014.
12. S. Bai, S. D. Galbraith, L. Li, and D. Sheffield. Improved combinatorial algorithms for the inhomogeneous short integer solution problem. *J. Cryptology*, 32(1):35–83, 2019.

13. S. Bai, D. Stehlé, and W. Wen. Improved reduction from the bounded distance decoding problem to the unique shortest vector problem in lattices. In *Proc. of ICALP*, pages 76:1–76:12, 2016.
14. S. Bai, D. Stehlé, and W. Wen. Measuring, simulating and exploiting the head concavity phenomenon in BKZ. In T. Peyrin and S. Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 369–404. Springer, Heidelberg, Dec. 2018.
15. A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 719–737. Springer, 2012.
16. A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *32nd Annual ACM Symposium on Theory of Computing*, pages 435–440. ACM Press, May 2000.
17. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584. ACM Press, June 2013.
18. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In *Proc. of CRYPTO*, volume 6841 of *LNCS*, pages 505–524. Springer, 2011.
19. Y. Chen. *Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe*. PhD thesis, Université Paris Diderot, 2009.
20. Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In D. H. Lee and X. Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, Heidelberg, Dec. 2011.
21. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In *Proc. of CRYPTO*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013.
22. U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Math. Comp.*, 44(170):463–471, 1985.
23. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In N. P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, Heidelberg, Apr. 2008.
24. N. Gama, P. Q. Nguyen, and O. Regev. Lattice enumeration using extreme pruning. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 257–278. Springer, Heidelberg, May / June 2010.
25. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008.
26. G. Hanrot, X. Pujol, and D. Stehlé. Algorithms for the shortest and closest lattice vector problems. In *IWCC*, volume 6639 of *LNCS*, pages 159–190. Springer, 2011.
27. G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 447–464. Springer, Heidelberg, Aug. 2011.
28. G. Herold, E. Kirshanova, and A. May. On the asymptotic complexity of solving lwe. *Des. Codes Cryptography*, 86(1):55–83, Jan. 2018.
29. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *15th Annual ACM Symposium on Theory of Computing*, pages 193–206. ACM Press, Apr. 1983.

30. R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.
31. S. Khot. Hardness of approximating the shortest vector problem in high L_p norms. In *Proc. of FOCS*, pages 290–297. IEEE Computer Society Press, 2003.
32. P. Kirchner and P.-A. Fouque. An improved BKW algorithm for LWE with applications to cryptography and lattices. Cryptology ePrint Archive, Report 2015/552, 2015. http://eprint.iacr.org/2015/552.
33. R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In A. Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, Heidelberg, Feb. 2011.
34. M. Liu and P. Q. Nguyen. Solving BDD by enumeration: An update. In E. Dawson, editor, *Topics in Cryptology – CT-RSA 2013*, volume 7779 of *Lecture Notes in Computer Science*, pages 293–309. Springer, Heidelberg, Feb. / Mar. 2013.
35. V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, Heidelberg, Apr. 2012.
36. V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *Proc. of CRYPTO*, pages 577–594, 2009.
37. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, Heidelberg, May / June 2010.
38. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381. IEEE Computer Society Press, Oct. 2004.
39. D. Micciancio and M. Walter. Fast lattice point enumeration with minimal overhead. In P. Indyk, editor, *26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 276–294. ACM-SIAM, Jan. 2015.
40. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005.
41. O. Regev. Lattice-based cryptography (invited talk). In C. Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 131–141. Springer, Heidelberg, Aug. 2006.
42. C. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2-3):201–224, 1987.
43. C. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. In *Proc. 8th International Symposium on Fundamentals of Computation Theory (FCT)*, pages 68–85, 1991.
44. C. P. Schnorr. *A more efficient algorithm for lattice basis reduction*, pages 359–369. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.
45. C. P. Schnorr. Lattice reduction by random sampling and birthday methods. In *Proceedings of the annual symposium on theoretical aspects of computer science (STACS 2003)*, volume 2607 of *LNCS*, pages 145–156. Springer, 2003.
46. The FPLLL development team. fplll, a lattice reduction library. Available at https://github.com/fplll/fplll, 2019.
47. The FPYLLL development team. fpylll, a python wrapper for fplll. Available at https://github.com/fplll/fplll, 2019.

48. Y. Wang, Y. Aono, and T. Takagi. Hardness evaluation for search lwe problem using progressive bkz simulator. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 101(12):2162–2170, 2018.