Recent improvements in concrete (quantum) cryptanalysis of some lattice problems

Eamonn W. Postlethwaite

Information Security Group, Royal Holloway, University of London

1st December 2020, York Seminar

Based on joint work with Martin R. Albrecht (who I thank for a lot of the experimental figures in this talk), Vlad Gheorghiu, John M. Schanck and Fernando Virdia.

Learning With Errors

Given $(\mathbf{A}, \mathbf{c})$ of the following form, find $\mathbf{s}$.

$$\begin{pmatrix} \\ \\ \mathbf{c} \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \leftarrow & n & \rightarrow \\ & & \\ & \mathbf{A} & \\ & & \\ & & \end{pmatrix} \cdot \begin{pmatrix} \\ \mathbf{s} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{e} \\ \\ \\ \end{pmatrix} \bmod q$$

Here $\mathbf{A} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{m \times n}\right)$, $\mathbf{s} \leftarrow \chi_s^n$, $\mathbf{e} \leftarrow \chi_e^m$, and $\mathbf{c} \in \mathbb{Z}_q^m$.

Learning With Errors

Given $(\mathsf{A}, \mathbf{c})$ of the following form, find $\mathbf{s}$.

$$\begin{pmatrix} \\ \\ \mathbf{c} \\ \\ \\ \end{pmatrix} = \begin{pmatrix} \leftarrow & n & \rightarrow \\ & & \\ & \mathsf{A} & \\ & & \\ & & \end{pmatrix} \cdot \begin{pmatrix} \\ \mathbf{s} \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \mathbf{e} \\ \\ \\ \end{pmatrix} \bmod q$$

Here $\mathsf{A} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{m \times n}\right), \mathbf{s} \leftarrow \chi_s^n, \mathbf{e} \leftarrow \chi_e^m$, and $\mathbf{c} \in \mathbb{Z}_q^m$.

Wider secret and error distributions (relative to q) give more *security* but less *functionality*.

Some facts about LWE

Not relevant for this talk but interesting,

- LWE has built public key encryption, key encapsulation, digital signatures, fully homomorphic encryption, non interactive zero knowledge for NP,
- there are reductions[1,2] from worst case lattice problems to LWE.

[1] Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *J. ACM* 56.6 (2009).
[2] Chris Peikert. "Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem: Extended Abstract". In: *STOC*. 2009, pp. 333–342.

Some facts about LWE

Not relevant for this talk but interesting,

- LWE has built public key encryption, key encapsulation, digital signatures, fully homomorphic encryption, non interactive zero knowledge for NP,
- there are reductions[1,2] from worst case lattice problems to LWE.

More relevant to this talk,

- originally the secret and error are drawn from the uniform distribution and the discrete Gaussian mod q respectively,
- there is a simple transformation that allows one to draw the secret from the same distribution as the error with (effectively) no loss: we call this *normal form*.

[1] Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *J. ACM* 56.6 (2009).
[2] Chris Peikert. "Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem: Extended Abstract". In: *STOC*. 2009, pp. 333–342.

A d dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^d$, and is described by a basis

$$\mathsf{B} = \begin{pmatrix} \uparrow & & \uparrow \\ \mathbf{b}_1 & \cdots & \mathbf{b}_r \\ \downarrow & & \downarrow \end{pmatrix} \in \mathbb{R}^{d \times r}, \quad \Lambda(\mathsf{B}) = \mathsf{B} \cdot \mathbb{Z}^r = \left\{ \sum_{i=1}^{r} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}.$$

The basis is formed of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_r \in \mathbb{R}^d$.

Introduction to lattices

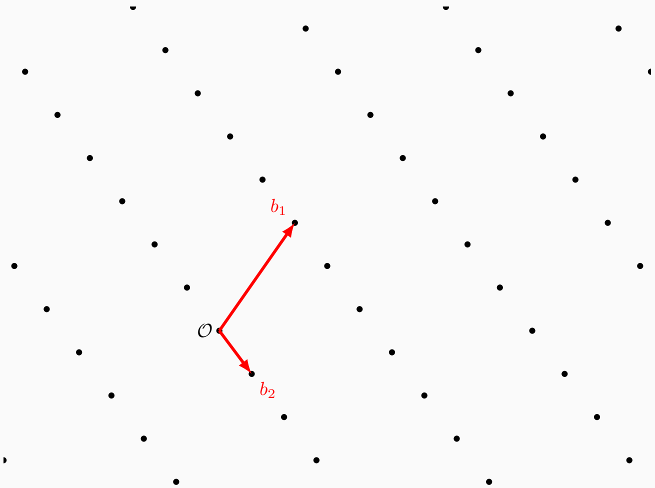
A d dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^d$, and is described by a basis

$$\mathsf{B} = \begin{pmatrix} \uparrow & & \uparrow \\ \mathbf{b}_1 & \cdots & \mathbf{b}_r \\ \downarrow & & \downarrow \end{pmatrix} \in \mathbb{R}^{d\times r}, \quad \Lambda(\mathsf{B}) = \mathsf{B}\cdot\mathbb{Z}^r = \left\{ \sum_{i=1}^{r} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}.$$

The basis is formed of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_r \in \mathbb{R}^d$.

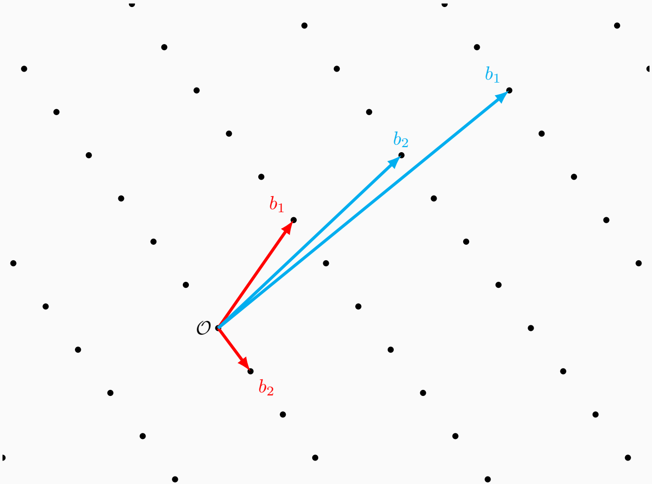
This is a rank r lattice, as its basis has r vectors in it, and any lattice with rank $r \geq 2$ will have infinitely many bases.

Some pictures



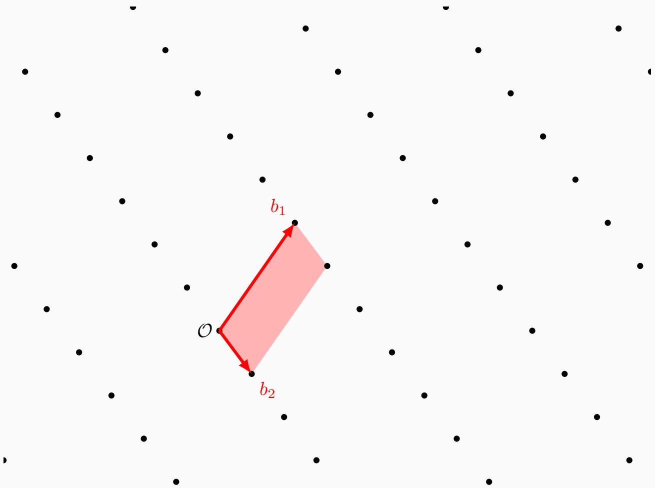
$\Lambda = \mathrm{span}_{\mathbb{Z}}(b_1, \ldots, b_r)$, $\mathbf{B} = \{b_1, \ldots, b_r\} \subset \mathbb{R}^d$ basis

Some pictures



Good basis **B**, bad basis **B**

Some pictures



The volume of the lattice $\mathrm{vol}(\Lambda)$ is an invariant (not dependent on e.g. basis $\mathbf{B}$).

How to attack LWE using the *primal* lattice

The idea is to construct a lattice basis using the $(\mathbf{A}, \mathbf{c})$ we get from the LWE problem.[3]

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_m & -\mathbf{A} & \mathbf{c} \\ 0 & \mathbf{I}_n & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{B} \cdot \begin{pmatrix} * \\ \mathbf{s} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ \mathbf{s} \\ 1 \end{pmatrix}, \quad d = r = m + n + 1.$$

This lattice has a *unique shortest vector* containing the error and secret!

[3] Shi Bai and Steven D. Galbraith. "Lattice Decoding Attacks on Binary LWE". In: *Information Security and Privacy*. 2014, pp. 322–337.

How to attack LWE using the *primal* lattice

The idea is to construct a lattice basis using the $(\mathbf{A}, \mathbf{c})$ we get from the LWE problem.[3]

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_m & -\mathbf{A} & \mathbf{c} \\ 0 & \mathbf{I}_n & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{B} \cdot \begin{pmatrix} * \\ \mathbf{s} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ \mathbf{s} \\ 1 \end{pmatrix}, \quad d = r = m + n + 1.$$

This lattice has a *unique shortest vector* containing the error and secret!

Use *lattice reduction* to solve uSVP.

[3] Shi Bai and Steven D. Galbraith. "Lattice Decoding Attacks on Binary LWE". In: *Information Security and Privacy*. 2014, pp. 322–337.

So what is lattice reduction?

An algorithm that takes as input a lattice basis, some parameters, and outputs a “better” basis for this lattice. We consider block Korkine–Zolotarev reduction, or BKZ.

So what is lattice reduction?

An algorithm that takes as input a lattice basis, some parameters, and outputs a “better” basis for this lattice. We consider block Korkine–Zolotarev reduction, or BKZ.

Ingredients of BKZ

- a lattice basis $\mathbf{B}$,
- a parameter called *blocksize*, $3 \leq \beta \leq r$,
- an SVP oracle O_{SVP} which returns a non zero shortest vector in some input lattice.

So what is lattice reduction?

An algorithm that takes as input a lattice basis, some parameters, and outputs a "better" basis for this lattice. We consider block Korkine–Zolotarev reduction, or BKZ.

Ingredients of BKZ

- a lattice basis $\mathbf{B}$,
- a parameter called *blocksize*, $3 \leq \beta \leq r$,
- an SVP oracle O_{SVP} which returns a non zero shortest vector in some input lattice.

Q: why not just use O_{SVP} on $\mathbf{B}$? A: the cost grows exponentially in the lattice rank.

So what is lattice reduction?

An algorithm that takes as input a lattice basis, some parameters, and outputs a “better” basis for this lattice. We consider block Korkine–Zolotarev reduction, or BKZ.

Ingredients of BKZ

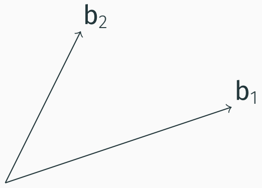
- a lattice basis $\mathbf{B}$,
- a parameter called *blocksize*, $3 \leq \beta \leq r$,
- an SVP oracle O_{SVP} which returns a non zero shortest vector in some input lattice.

Q: why not just use O_{SVP} on $\mathbf{B}$? A: the cost grows exponentially in the lattice rank.

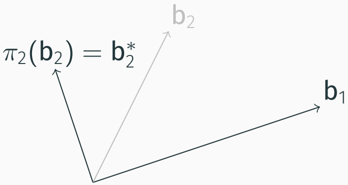
We call O_{SVP} on many related rank β lattices to find a shortish vector in $\Lambda(\mathbf{B})$.

BKZ I

We need a projection operator $\pi_{\mathbf{B},i} \colon \mathbb{R}^d \to \mathbb{R}^d$ for $1 \leq i \leq r$ that removes the components of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. Visually,



We need a projection operator $\pi_{\mathbf{B},i} \colon \mathbb{R}^d \to \mathbb{R}^d$ for $1 \leq i \leq r$ that removes the components of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. Visually,



Note

- $\pi_{\mathbf{B},1}$ is the identity (no projection),
- we remove the $\mathbf{B}$ from the subscript,
- $\pi_i(\mathbf{b}_i) = \mathbf{b}_i^*$, the *Gram–Schmidt* orthogonalisation of $\mathbf{b}_i$.

BKZ II

Data: lattice basis $\mathbf{B}$
Data: blocksize β
repeat *for* τ tours
    for $i \leftarrow 1$ **to** $r-1$ **do**
        the block begins at $\mathbf{b}_i$
        the block ends at $\mathbf{b}_f$ for $f = \min(i+\beta-1, r)$
        form *block* $\mathbf{B}_{[i:f]} = (\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_f))$ `// first vector is` $\mathbf{b}_i^*$
        $\mathbf{v} \leftarrow O_{SVP}(\mathbf{B}_{[i:f]})$
        insert $\mathbf{v}$ into $\mathbf{B}$
    end

BKZ III

$$\begin{pmatrix} \overbrace{\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3 \quad \mathbf{b}_4 \quad \mathbf{b}_5}^{\beta = 5} & \mathbf{b}_6 & \mathbf{b}_7 & \mathbf{b}_8 & \dots \end{pmatrix}$$

BKZ III

$$\begin{pmatrix} \overbrace{\underbrace{b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5}}^{\beta = 5} \quad b_6 \quad b_7 \quad b_8 \quad \dots \end{pmatrix}$$

BKZ III

$$\left(\begin{matrix} \overbrace{b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5}^{\beta = 5} & b_6 & b_7 & b_8 & \dots \end{matrix}\right)$$

BKZ III

$$\left(b_1 \quad \overbrace{b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6}^{\beta = 5} \quad b_7 \quad b_8 \quad \ldots \right)$$

BKZ III

$$\begin{pmatrix} \color{red}{b_1} & \overbrace{\underbrace{b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6}}^{\beta = 5} & b_7 & b_8 & \dots \end{pmatrix}$$

BKZ III

$$\left(\; \mathbf{b}_1 \quad \overbrace{\mathbf{b}_2 \quad \mathbf{b}_3 \quad \mathbf{b}_4 \quad \mathbf{b}_5 \quad \mathbf{b}_6}^{\beta = 5} \quad \mathbf{b}_7 \quad \mathbf{b}_8 \quad \ldots \; \right)$$

BKZ III

$$\begin{pmatrix} b_1 & b_2 & \overbrace{b_3 \quad b_4 \quad b_5 \quad b_6 \quad b_7}^{\beta = 5} & b_8 & \dots \end{pmatrix}$$

BKZ III

$$\begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \overbrace{\mathbf{b}_3 \quad \mathbf{b}_4 \quad \mathbf{b}_5 \quad \mathbf{b}_6 \quad \mathbf{b}_7}^{\beta = 5} & \mathbf{b}_8 & \dots \end{pmatrix}$$

BKZ III

$$\left(\mathbf{b}_1 \quad \mathbf{b}_2 \quad \overbrace{\mathbf{b}_3 \quad \mathbf{b}_4 \quad \mathbf{b}_5 \quad \mathbf{b}_6 \quad \mathbf{b}_7}^{\beta = 5} \quad \mathbf{b}_8 \quad \ldots \right)$$

Choose your own adventure

There are (at least) two natural questions to ask next.

- How well does BKZ perform in the primal attack $\leftrightsquigarrow$ how large must we take β?
- How expensive is BKZ for a given β $\leftrightsquigarrow$ how expensive is O_{SVP}?

The output of BKZ

On random lattices the average case behaviour of BKZ[4] with blocksize $\beta \geq 50$ is to output a basis $\mathbf{B}$ for the input lattice with

$$\|\mathbf{b}_1\| \approx \delta_\beta^{r-1} \cdot \mathrm{vol}\,(\Lambda)^{1/r}, \quad \delta_\beta = \left(\frac{\beta}{2\pi e} \cdot (\pi\beta)^{1/\beta} \right)^{1/(2(\beta-1))}.$$

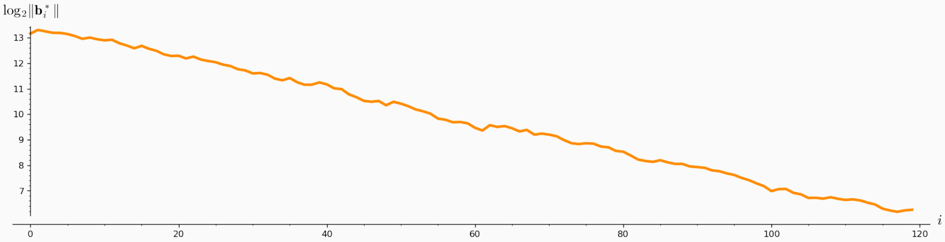
We recall the volume of a lattice from earlier, it can be computed as

$$\mathrm{vol}(\Lambda) = \prod_{i=1}^{r} \|\mathbf{b}_i^*\|.$$

[4] Yuanmi Chen. “Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe”. PhD thesis. Université Paris Diderot, 2013.

Geometric Series Assumption

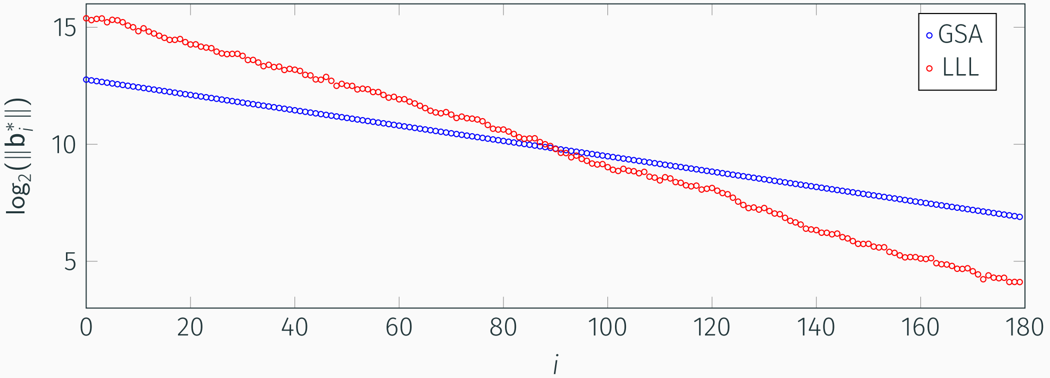
We plot the log norms of $\left\|\mathbf{b}_i^*\right\|$ against the index i.



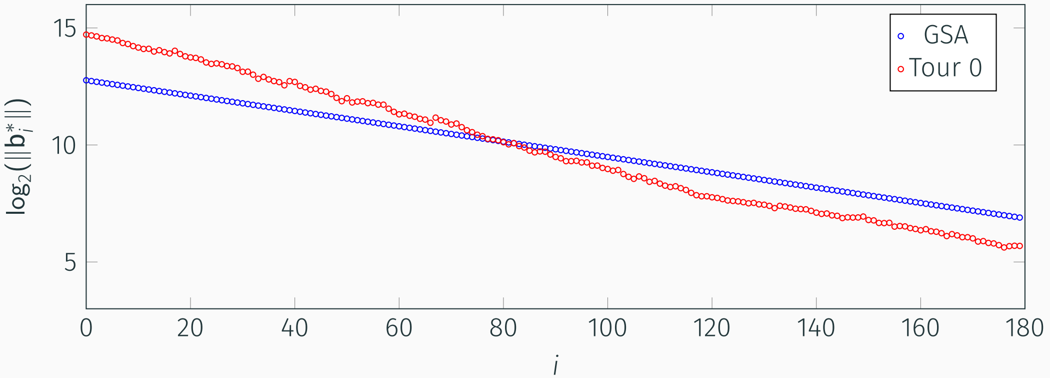
The GSA says stronger lattice reduction equals a flatter slope.[5]

[5] Claus Peter Schnorr. "Lattice Reduction by Random Sampling and Birthday Methods". In: *STACS 2003*. 2003, pp. 145–156.

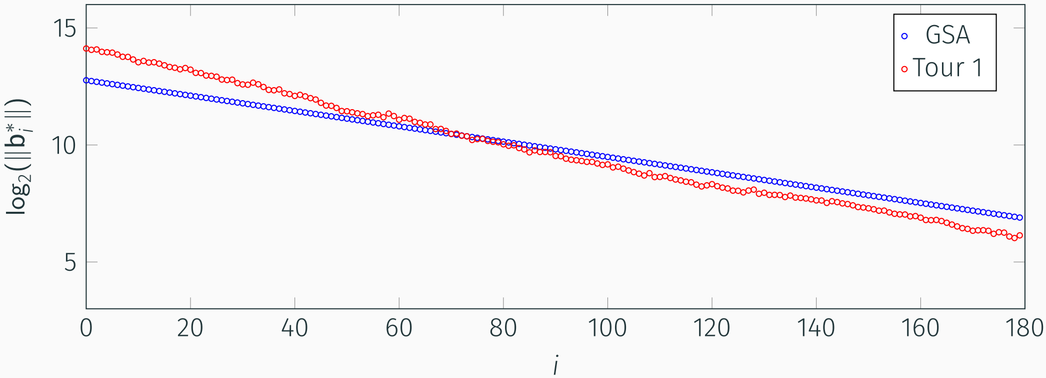
Behaviour in Practice: BKZ-60 in Dimension 180 i



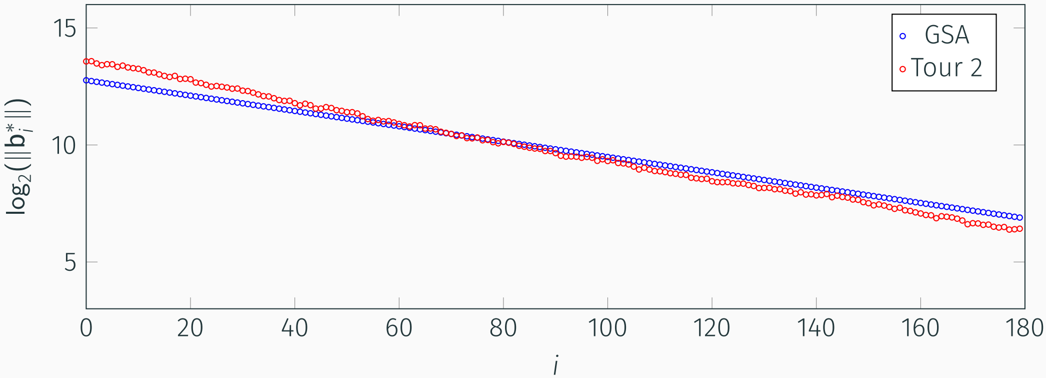
Behaviour in Practice: BKZ-60 in Dimension 180 ii



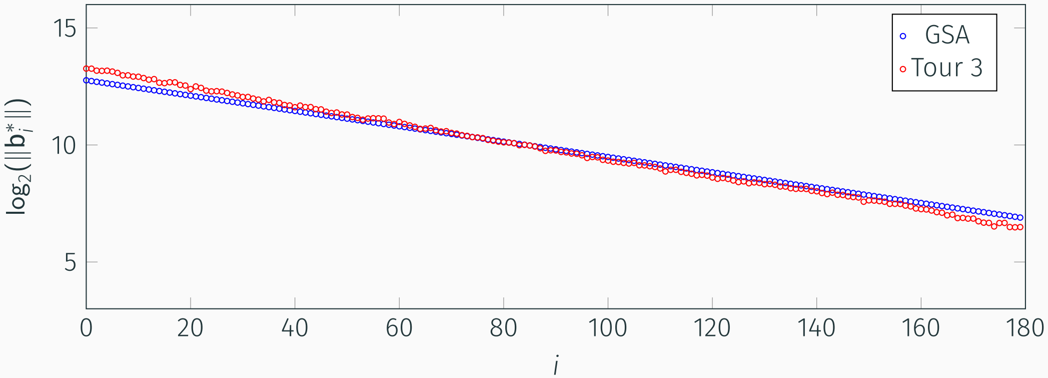
Behaviour in Practice: BKZ-60 in Dimension 180 iii



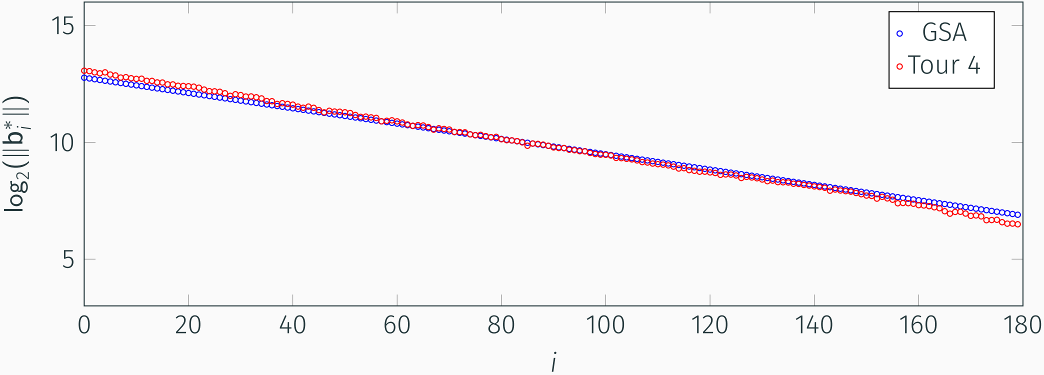
Behaviour in Practice: BKZ-60 in Dimension 180 iv



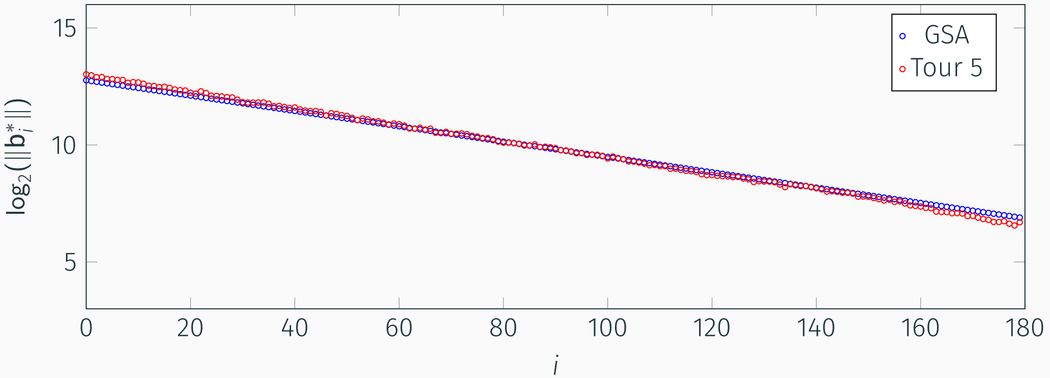
Behaviour in Practice: BKZ-60 in Dimension 180 v



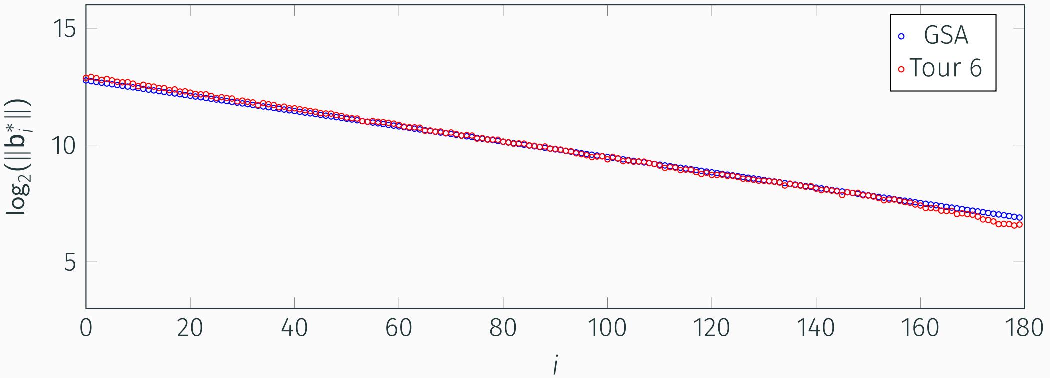
Behaviour in Practice: BKZ-60 in Dimension 180 vi



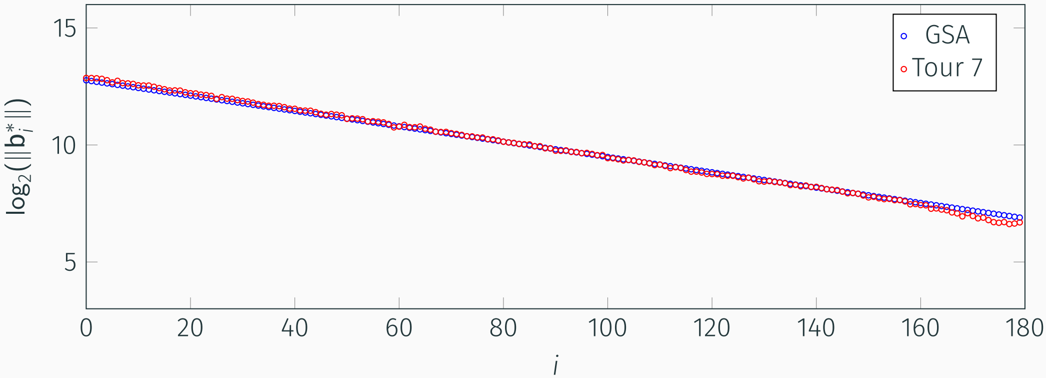
Behaviour in Practice: BKZ-60 in Dimension 180 vii



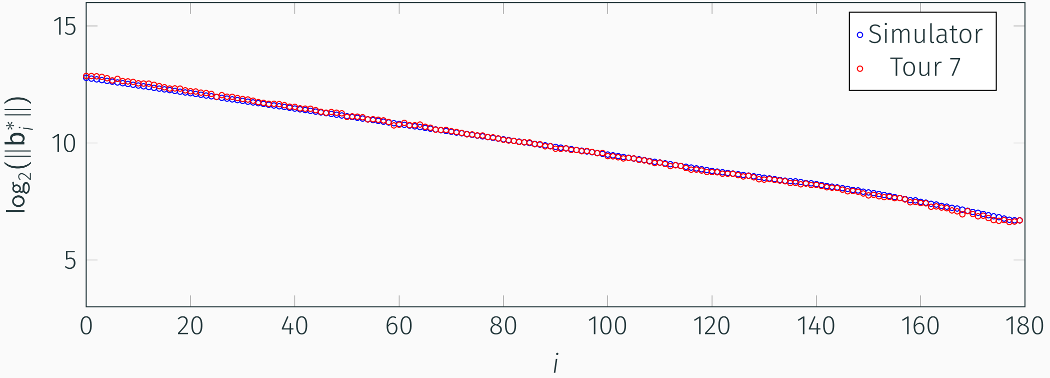
Behaviour in Practice: BKZ-60 in Dimension 180 viii



Behaviour in Practice: BKZ-60 in Dimension 180 ix



Behaviour in Practice: BKZ-60 in Dimension 180 x



[6]

[6] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates". In: *ASIACRYPT*. 2011, pp. 1–20.

What about the uSVP solution?

Consider normal form LWE instances, so $\mathbf{s} \leftarrow \chi^n, \mathbf{e} \leftarrow \chi^m$ where $\chi = \chi_e = \chi_s$. Also consider χ such that $\mathbb{E}(\chi) = 0$ and $\mathbb{V}(\chi) = \sigma^2$ for some σ, so that $\mathbb{E}(\chi^2) = \mathbb{V}(\chi) = \sigma^2$.

Name	χ	$\mathbb{E}(\chi)$	$\mathbb{V}(\chi)$
discrete Gaussian mod q	$D_{q,\sigma}$	0	σ^2
centred binary	$\mathcal{U}(\{-1, 1\})$	0	1
trinary	$\mathcal{U}(\{-1, 0, 1\})$	0	$2/3$
bounded uniform	$\mathcal{U}(\{-B, \ldots, 0, \ldots, B\})$	0	$B(B+1)/3$

The uSVP solution is a vector $\mathbf{t} = (\mathbf{e} \,|\, \mathbf{s} \,|\, 1)^T$.

Since $d = m + n + 1$, then $\mathbf{t}$ has expected square length $m\sigma^2 + n\sigma^2 + 1^2 \approx d\sigma^2$.

What about the uSVP solution when projected?

We model the squared length of the projections of $\mathbf{t}$ using a *chi-squared* distribution

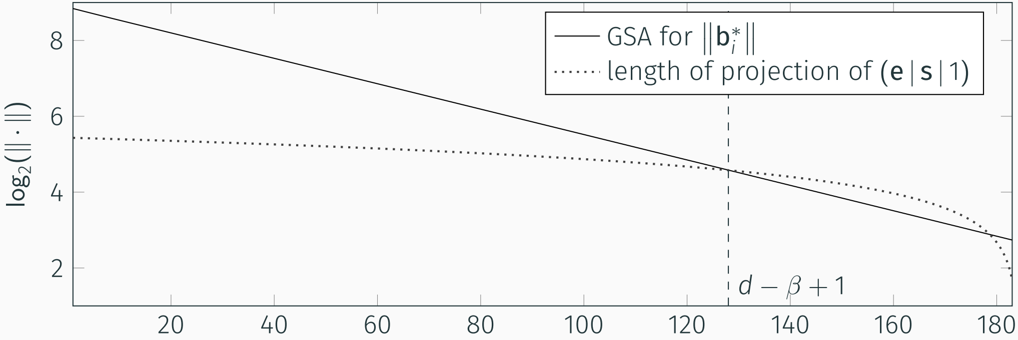
$$\|\pi_i(\mathbf{t})\|^2 \sim \sigma^2 \cdot \chi^2(d - i + 1).$$

This gives us the expected squared length of the projection of $\mathbf{t}$ in a block in BKZ!

In particular, for some block $\mathbf{B}_{[i:\, i+\beta-1]} = (\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_{i+\beta-1}))$ we model the expected square length of the projection as $(d - i + 1)\sigma^2$.

Now we have all the pieces...

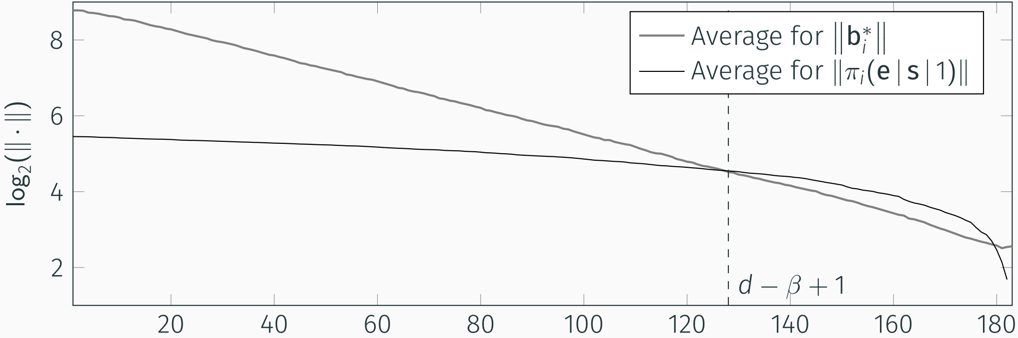
In the final block of a tour, if the projection of $\mathbf{t}$ is the shortest vector, then the O_{SVP} subroutine of BKZ will find it. Pictorially[7]



[7] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. “Revisiting the Expected Cost of Solving uSVP and Applications to LWE”. In: *ASIACRYPT*. 2017, pp. 297–322.

Now we have all the pieces...

In the final block of a tour, if the projection of $\mathbf{t}$ is the shortest vector, then the O_{SVP} subroutine of BKZ will find it. Pictorially[7]



[7] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. "Revisiting the Expected Cost of Solving uSVP and Applications to LWE". In: *ASIACRYPT*. 2017, pp. 297–322.

The NewHope approach

This method for determining the required β to recover $\mathbf{t}$ was introduced in a lattice KEM paper called NewHope.[8]

$$\sigma\sqrt{\beta} \leq \delta_\beta^{2\beta-d-1} \cdot \mathrm{vol}\left(\Lambda\right)^{1/d} = \delta_\beta^{2\beta-d-1} \cdot q^{m/d}.$$

Experimental works vindicated the approach,[9,10] but also noticed that smaller β sometimes had a non zero chance of success.

[8] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. "Post-quantum Key Exchange—A New Hope". In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 327–343.

[9] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. "Revisiting the Expected Cost of Solving uSVP and Applications to LWE". In: *ASIACRYPT*. 2017, pp. 297–322.

[10] Shi Bai, Shaun Miller, and Weiqiang Wen. "A Refined Analysis of the Cost for Solving LWE via uSVP". In: *AFRICACRYPT*. 2019, pp. 181–205.

An example

Adapted from[11] – averaged over 500 trials.

n	q	σ	β_{2016}	m_{2016}	β	% success
100	2053	$8/\sqrt{2\pi}$	67	243	67	88.8
					62	39.6
					57	5.8
					52	0.2

[11] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. "Revisiting the Expected Cost of Solving uSVP and Applications to LWE". In: *ASIACRYPT*. 2017, pp. 297–322.

On the Success Probability of Solving Unique SVP via BKZ[14]

In recent work with Fernando Virdia[12] we use ideas from[13] to explain, estimate and experiment on these instances where a smaller β than expected can sometimes succeed.

The crucial idea is that, since we are modelling

$$\|\pi_i(\mathbf{t})\|^2 \sim \sigma^2 \cdot \chi^2(d - i + 1),$$

we can argue about the probabilities that projections of $\mathbf{t}$ have a given length, rather than relying on the expectation of these lengths.

[12] Find him here `https://fundamental.domains/`

[13] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. "LWE with Side Information: Attacks and Concrete Security Estimation". In: *CRYPTO*. 2020, pp. 329–358.

[14] Eamonn W. Postlethwaite and Fernando Virdia. *On the Success Probability of Solving Unique SVP via BKZ*. Cryptology ePrint Archive, Report 2020/1308. `https://eprint.iacr.org/2020/1308`. 2020.

A uSVP simulator for BKZ

Input: (n, q, χ, m), β, τ
$p_{\text{tot}} \leftarrow 0, \sigma^2 \leftarrow \mathbb{V}(\chi)$
$d \leftarrow n + m + 1$
for $tour \leftarrow 1$ **to** τ **do**
  $\texttt{profile} \leftarrow \text{BKZSim}((n, q, \chi, m), \beta, \text{tour})$
  $p_{\text{new}} \leftarrow P[x \leftarrow \sigma^2\chi^2(\beta) : x \leq \texttt{profile}[d-\beta+1]]$
  $p_{\text{tot}} \leftarrow p_{\text{tot}} + (1 - p_{\text{tot}}) \cdot p_{\text{new}}$
end
return p_{tot}

We are assuming the independence of tours; the insertions and extra processing “rerandomise” the basis.

We also describe a uSVP simulator for a variant called *progressive* BKZ, where the blocksize increments after τ tours.

BKZ experiments

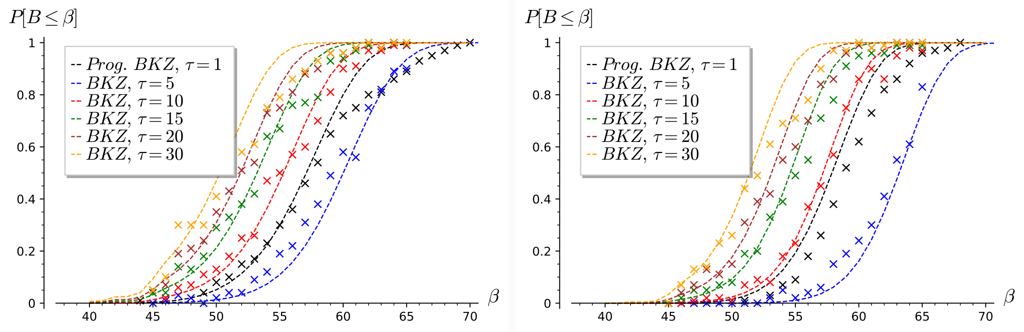


Figure 1: Left: $(n, q, \sigma) = (72, 97, 1)$. Right: $(n, q, \sigma) = (100, 257, \sqrt{2/3})$. Both: $\beta_{2016} \approx 60$ and using discrete Gaussian $D_{q,\sigma}$ for secret and error.

Progressive BKZ experiments

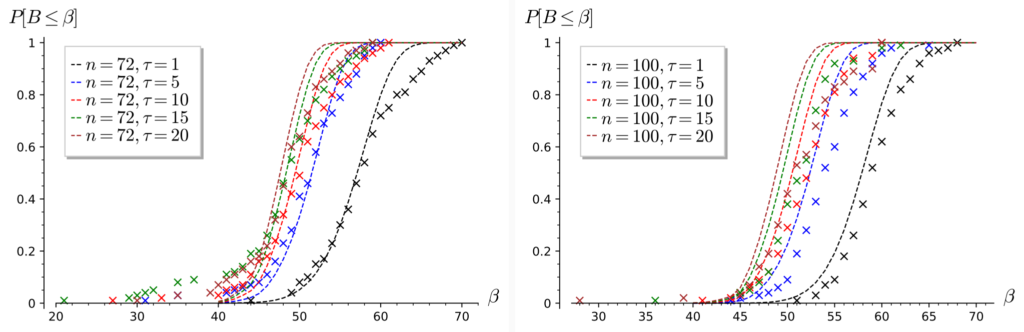


Figure 2: Left: $(n, q, \sigma) = (72, 97, 1)$. Right: $(n, q, \sigma) = (100, 257, \sqrt{2/3})$. Both: $\beta_{2016} \approx 60$ and using discrete Gaussian $D_{q,\sigma}$ for secret and error.

Accuracy?

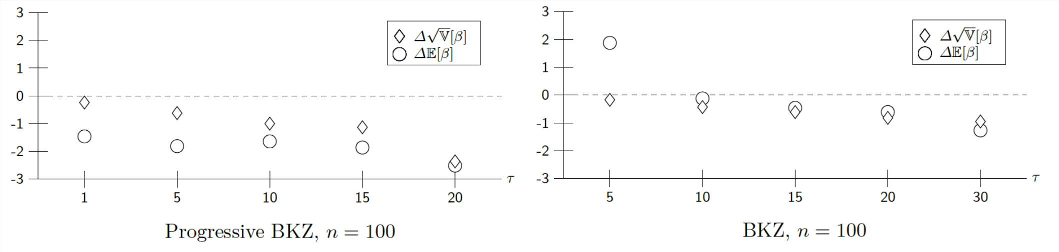


Figure 3: Difference in expectation and variance between experimental and simulated worlds.

Rule of thumb: a difference of 3 in blocksize is approximately 1 bit of security.

The independence condition breaks down when little further improvement can be made to the basis; this happens sooner for progressive BKZ.

Findings I

It is the variance of the distribution from which **e** and **s** are drawn that determines the concrete complexity of the primal attack.

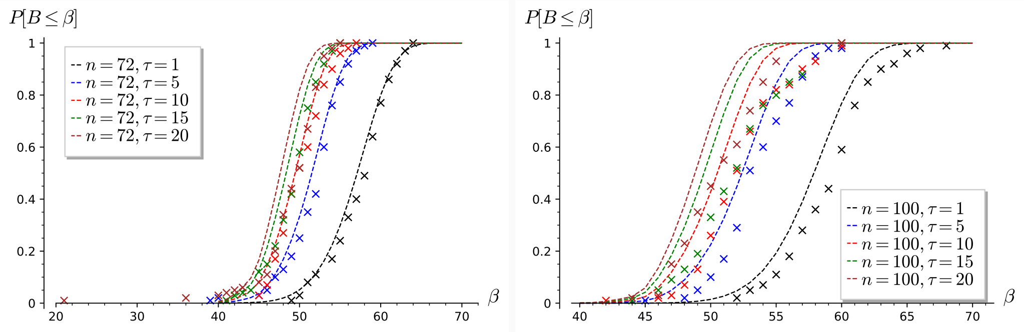


Figure 4: Left: $(n, q, \sigma) = (72, 97, 1)$, centred binary secret and error. Right: $(n, q, \sigma) = (100, 257, \sqrt{2/3})$, ternary secret and error. Both: $\beta_{2016} \approx 60$.

Findings II

The expected sample variance plays a role in how accurate our simulators are. Considering $\mathbf{t} = (\mathbf{t}_1, \ldots, \mathbf{t}_d)$ the sample variance is

$$s^2 = \frac{1}{d}\sum_{i=1}^{d}(\mathbf{t}_i - t)^2, \quad t = \frac{1}{d}\sum_{i=1}^{d}\mathbf{t}_i.$$

Depending on the value of s^2, then the model $\|\pi_i(\mathbf{t})\|^2 \sim \sigma^2 \cdot \chi^2(d - i + 1)$ may be inaccurate.

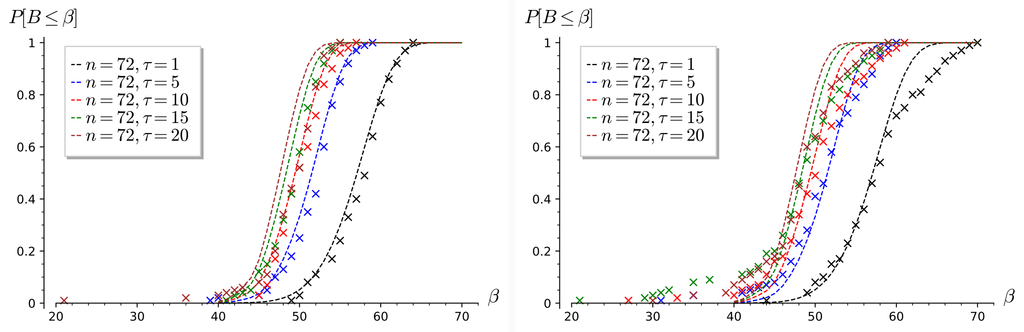


Figure 5: Left: secret and error from centred binary. Right: secret and error from discrete Gaussian. Both: $(n, q, \sigma) = (72, 97, 1)$, $\beta_{2016} \approx 60$.

Findings II

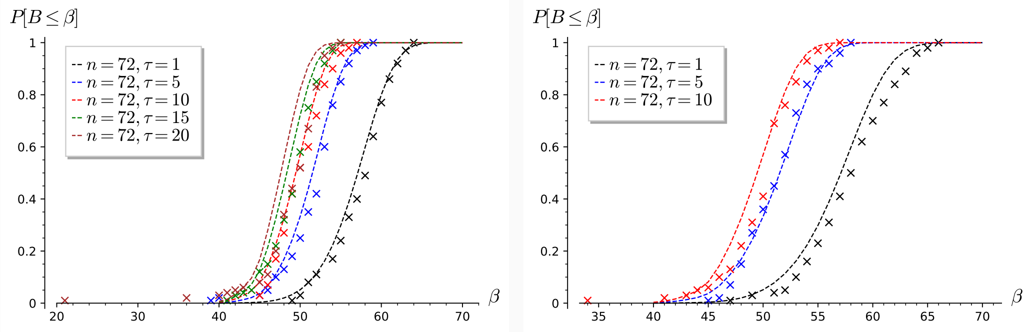


Figure 5: Left: secret and error from centred binary. Right: secret and error from discrete Gaussian, corrected sample variance. Both: $(n, q, \sigma) = (72, 97, 1)$, $\beta_{2016} \approx 60$.

Applied to a NIST candidate[15]

scheme	n	q	σ_s	σ_e	β_{2016}	m_{2016}	BKZ, $\tau = 15$ $\mathbb{E}(\text{succ. } \beta)$	BKZ, $\tau = 15$ $\sqrt{\mathbb{V}}(\text{succ. } \beta)$
Kyber 512	512	3329	1	1	381	484	386.06	2.56
Kyber 768	768	3329	1	1	623	681	634.41	2.96
Kyber 1024	1024	3329	1	1	873	860	891.13	3.31

Progressive BKZ, $\tau = 1$		Progressive BKZ, $\tau = 5$	
$\mathbb{E}(\text{succ. } \beta)$	$\sqrt{\mathbb{V}}(\text{succ. } \beta)$	$\mathbb{E}(\text{succ. } \beta)$	$\sqrt{\mathbb{V}}(\text{succ. } \beta)$
389.53	2.88	385.70	2.32
638.23	3.30	634.00	2.66
895.24	3.66	890.63	2.96

[15] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. *CRYSTALS-KYBER*. Tech. rep. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`. National Institute of Standards and Technology, 2019.

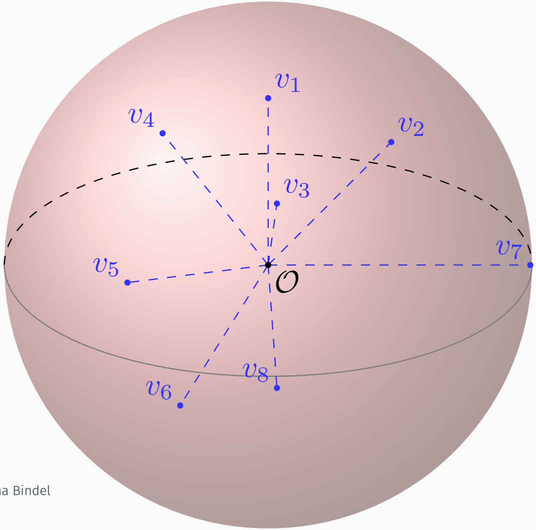
Briefly: lattice sieving

Lattice sieves take a basis of lattice and output an (approximate solution) to the shortest vector problem. They have complexity

$$\text{Time: } 2^{\Theta(d)} \qquad \text{Space: } 2^{\Theta(d)}$$

They are one way to instantiate the O_{SVP} oracle within BKZ.

They key subroutine is nearest neighbour search on the sphere



find pairs (v_i, v_j) such that $\|v_i - v_j\| \leq 1 \iff \langle v_i, v_j \rangle \geq \cos(\pi/3)$.

Set the problem up as a search predicate

Let $[N] = \{1, \ldots, N\}$ and $f\colon [N] \to \{0, 1\}$ be an unstructured predicate, with *roots*

$$\mathsf{Ker}(f) = \{x\colon f(x) = 0\}.$$

Set the problem up as a search predicate

Let $[N] = \{1, \ldots, N\}$ and $f \colon [N] \to \{0, 1\}$ be an unstructured predicate, with *roots*

$$\mathrm{Ker}(f) = \{x \colon f(x) = 0\}.$$

We can find a root

- classically by evaluating $f(1), \ldots, f(N)$,
- quantumly by measuring $\mathbf{G}(f)^j \mathbf{D}|0\rangle$.

Set the problem up as a search predicate

Let $[N] = \{1, \dots, N\}$ and $f \colon [N] \to \{0, 1\}$ be an unstructured predicate, with *roots*

$$\mathsf{Ker}(f) = \{x \colon f(x) = 0\}.$$

We can find a root

- classically by evaluating $f(1), \dots, f(N)$,
- quantumly by measuring $\mathbf{G}(f)^j \mathbf{D}|0\rangle$.

If $|\mathsf{Ker}(f)| \ll N$ then we expect $O(N)$ queries to f classically, and $j \in O(\sqrt{N})$ queries to $\mathbf{G}(f)$ quantumly.

Use a filter

A potentially cheaper way is to use a filter, some predicate

$$g\colon [N] \to \{0,1\}, |\mathrm{Ker}(g) \cap \mathrm{Ker}(f)| \geq 1.$$

Then (classically) we can evaluate

$$g(1), f(1) \text{ when } g(1) = 0, \ldots, g(N), f(N) \text{ when } g(N) = 0.$$

Use a filter

A potentially cheaper way is to use a filter, some predicate

$$g\colon [N] \to \{0,1\}, |\mathrm{Ker}(g) \cap \mathrm{Ker}(f)| \geq 1.$$

Then (classically) we can evaluate

$$g(1), f(1) \text{ when } g(1) = 0, \ldots, g(N), f(N) \text{ when } g(N) = 0.$$

What makes a good filter? Cheap to evaluate, and

$$\rho_f(g) = 1 - \frac{|\mathrm{Ker}(f) \cap \mathrm{Ker}(g)|}{|\mathrm{Ker}(g)|}, \quad \eta_f(g) = 1 - \frac{|\mathrm{Ker}(f) \cap \mathrm{Ker}(g)|}{|\mathrm{Ker}(f)|}$$

the false positive and negative rate, are both small.

A filtered quantum search

Branching based on g is not possible within Grover's algorithm. We give a technical lemma that (roughly) says, let

A filtered quantum search

Branching based on g is not possible within Grover's algorithm. We give a technical lemma that (roughly) says, let

- g be a filter for predicate $f : [N] \to \{0, 1\}$,
- $P, Q, \gamma \in \mathbb{R}$ such that
  - $P/\gamma \leq |\mathsf{Ker}(g)| \leq \gamma P$, and
  - $1 \leq Q \leq |\mathsf{Ker}(f) \cap \mathsf{Ker}(g)|$.

Then we can find a root of f with constant probability with a cost dominated by $\frac{\gamma}{2}\sqrt{N/Q}$ calls to $\mathsf{G}(g)$.

A filtered quantum search

Branching based on g is not possible within Grover's algorithm. We give a technical lemma that (roughly) says, let

- g be a filter for predicate $f : [N] \to \{0, 1\}$,
- $P, Q, \gamma \in \mathbb{R}$ such that
  - $P/\gamma \leq |\mathsf{Ker}(g)| \leq \gamma P$, and
  - $1 \leq Q \leq |\mathsf{Ker}(f) \cap \mathsf{Ker}(g)|$.

Then we can find a root of f with constant probability with a cost dominated by $\frac{\gamma}{2}\sqrt{N/Q}$ calls to $\mathsf{G}(g)$.

The cost of a Grover query encoding the filter, $\mathsf{G}(g)$, and not one encoding the predicate, $\mathsf{G}(f)$, is then the crucial quantity.

What is the filter?

For lattices vectors $u, v_1, \ldots, v_N$, the reduction predicate of u is

$$f_u \colon \{v_1, \ldots, v_N\} \to \{0,1\},\ f_u(v_i) = 0 \iff \langle u, v_i \rangle \geq \cos(\pi/3).$$

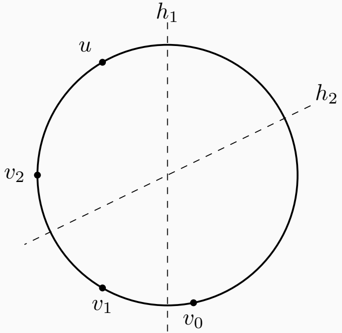
[16] Robert Fitzpatrick, Christian Bischof, Johannes Buchmann, Özgür Dagdelen, Florian Göpfert, Artur Mariano, and Bo-Yin Yang. "Tuning GaussSieve for Speed". In: *LATINCRYPT*. 2014.

What is the filter?

For lattices vectors $u, v_1, \ldots, v_N$, the reduction predicate of u is

$$f_u \colon \{v_1, \ldots, v_N\} \to \{0, 1\},\ f_u(v_i) = 0 \iff \langle u, v_i \rangle \geq \cos(\pi/3).$$

For the filter g we use 'XOR and popcount'.[16]



$$\texttt{popcount}_{1,2}(u, v_0) = 1$$

$$\texttt{popcount}_{1,2}(u, v_1) = 0$$

$$\texttt{popcount}_{1,2}(u, v_2) = 0$$

$$(k, n) = (1, 2)$$

[16] Robert Fitzpatrick, Christian Bischof, Johannes Buchmann, Özgür Dagdelen, Florian Göpfert, Artur Mariano, and Bo-Yin Yang. "Tuning GaussSieve for Speed". In: *LATINCRYPT*. 2014.

Ingredients

Putting it together

- we have a filtered quantum search routine,
- we have a filter, `popcount`, and build an optmised quantum circuit for it,
- we give an analysis of the false positive and negative rates of `popcount`
- we define a number of metrics depending on assumptions regarding quantum memory.

A selected result

Our estimates suggest less than advantage for quantum sieves than the asymptotics suggest, without entirely ruling out their relevance.

Quantum Metric	d	$\log_2$ time$_C$	$\log_2$ time$_Q$	$\log_2$ memory	$0.0272d$
GE19[17]	312	119	119	78	8.5
GE19	352	130	128	87	9.6
GE19	824	270	256	187	22.4
GE19	544	189	182	128	14.8

All classical costs are in a simple RAM model, the above table is for ListDecodingSieve.[18]

[17] Craig Gidney and Martin Ekerå. *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits.* 2019. arXiv: `1905.09749 [quant-ph]`.

[18] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. “New Directions in Nearest Neighbor Searching with Applications to Lattice Sieving”. In: *SODA*. 2016.

Conclusion

Even given the following

- we cost qRAM and RAM as the same (unit cost),
- we are conservative within our filtered quantum search,
- we do not consider depth constraints, which harm quantum search more,

we see a smaller quantum advantage than expected.

Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. “Post-quantum Key Exchange—A New Hope”. In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 327–343.

Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. “Revisiting the Expected Cost of Solving uSVP and Applications to LWE”. In: *ASIACRYPT*. 2017, pp. 297–322.

Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. “New Directions in Nearest Neighbor Searching with Applications to Lattice Sieving”. In: *SODA*. 2016.

Shi Bai and Steven D. Galbraith. “Lattice Decoding Attacks on Binary LWE”. In: *Information Security and Privacy*. 2014, pp. 322–337.

References ii

Shi Bai, Shaun Miller, and Weiqiang Wen. “A Refined Analysis of the Cost for Solving LWE via uSVP”. In: *AFRICACRYPT*. 2019, pp. 181–205.

Yuanmi Chen. “Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe”. PhD thesis. Université Paris Diderot, 2013.

Yuanmi Chen and Phong Q. Nguyen. “BKZ 2.0: Better Lattice Security Estimates”. In: *ASIACRYPT*. 2011, pp. 1–20.

Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. “LWE with Side Information: Attacks and Concrete Security Estimation”. In: *CRYPTO*. 2020, pp. 329–358.

Robert Fitzpatrick, Christian Bischof, Johannes Buchmann, Özgür Dagdelen, Florian Göpfert, Artur Mariano, and Bo-Yin Yang. “Tuning GaussSieve for Speed”. In: *LATINCRYPT*. 2014.

Craig Gidney and Martin Ekerå. *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits*. 2019. arXiv: `1905.09749 [quant-ph]`.

Chris Peikert. "Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem: Extended Abstract". In: *STOC*. 2009, pp. 333–342.

Eamonn W. Postlethwaite and Fernando Virdia. *On the Success Probability of Solving Unique SVP via BKZ*. Cryptology ePrint Archive, Report 2020/1308. `https://eprint.iacr.org/2020/1308`. 2020.

Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *J. ACM* 56.6 (2009).

References iv

Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. *CRYSTALS-KYBER*. Tech. rep. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`. National Institute of Standards and Technology, 2019.

Claus Peter Schnorr. “Lattice Reduction by Random Sampling and Birthday Methods”. In: *STACS 2003*. 2003, pp. 145–156.