Estimating the Difficulty of Breaking Lattice-Based Cryptography

with a focus on LWE

Martin R. Albrecht

King's College London & SandboxAQ

The Plan

- Introduce estimating the cost of solving LWE-based schemes
  - strategies (primal and dual lattice attacks)
  - lattice reduction
  - finding short vectors
  - ML-based attacks
- Introduce usage of ”lattice estimator” along the way

`https://github.com/malb/lattice-estimator/`

Ack

First: A big shout out to Ben Curtis, who is the reason why the estimator remains relevant to this community. He's the one fixing performance and correctness issues encountered for FHE parameters!

Example

```
from estimator import *
from estimator.schemes import Kyber768
_ = LWE.estimate.rough(Kyber768)
```

```
usvp                 :: rop: ≈2^182.2, red: ≈2^182.2, δ: 1.002902, β: 624, d: 1427, tag: usvp
dual_hybrid          :: rop: ≈2^173.7, red: ≈2^173.4, guess: ≈2^171.1, β: 594, p: 4, ζ: 5, t: 70, β': 594, …
```

```
_ = LWE.estimate(Kyber768)
```

```
bkw                  :: rop: ≈2^238.3, m: ≈2^225.5, mem: ≈2^226.5, b: 19, t1: 1, t2: 17, ℓ: 18, …
usvp                 :: rop: ≈2^204.9, red: ≈2^204.9, δ: 1.002902, β: 624, d: 1427, tag: usvp
bdd                  :: rop: ≈2^201.0, red: ≈2^200.0, svp: ≈2^200.0, β: 606, η: 641, d: 1425, …
dual                 :: rop: ≈2^214.2, mem: ≈2^142.8, m: 723, β: 653, d: 1491, ↻: 1, tag: dual
dual_hybrid          :: rop: ≈2^196.1, red: ≈2^195.5, guess: ≈2^194.4, β: 586, p: 4, ζ: 15, …
```

Computational Problems

Given $(\mathbf{A}, \mathbf{c})$, find $\mathbf{s}$ when

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

for $\mathbf{c} \in \mathbb{Z}_q^m$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, and $\mathbf{s} \in \mathbb{Z}^n$ and $\mathbf{e} \in \mathbb{Z}^m$ having small entries.

Learning with Errors

Given $(\mathbf{A}, \mathbf{c})$, find $\mathbf{s}$ when

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

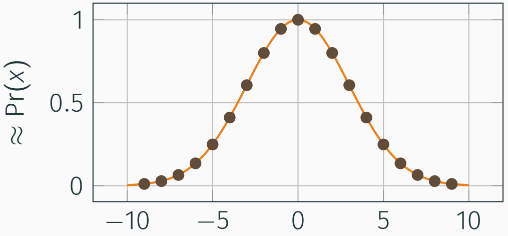
for $\mathbf{c} \in \mathbb{Z}_q^m$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, and $\mathbf{s} \in \mathbb{Z}^n$ and $\mathbf{e} \in \mathbb{Z}^m$ having small entries.

"Having Small Entries"

Discrete Gaussian the classic, annoying to sample from; ➡

Binomial $\sum_{0 \leq i < \eta} (b_i - b_{\eta+i}) \mid b_j \leftarrow\$ \{0, 1\}$

Small Uniform $\leftarrow\$ [a, \ldots, b] \mid a, b \in \mathbb{N}$




```
from estimator import *
ND.DiscreteGaussian(stddev=2), ND.CenteredBinomial(eta=8), ND.Uniform(-3, 3)
```

```
(D(σ=2.00), D(σ=2.00), D(σ=2.00))
```

- Literature assumes these all behave essentially the same under attacks
- No loss in security if secret **s** and error **e** have same distribution [ACPS09]

Estimator

```
Kyber768 = LWEParameters(
    n=3 * 256,
    q=3329,
    Xs=ND.CenteredBinomial(2),
    Xe=ND.CenteredBinomial(2),
    m=3 * 256,
    tag="Kyber 768",
)
```

Ad Break: We now do NTRU, too!

```
Falcon512_SKR = NTRU.Parameters(
    n=512,
    q=12289,
    Xs=ND.DiscreteGaussian(4.0532),
    Xe=ND.DiscreteGaussian(4.0532),
    m=512,
    ntru_type="circulant",
    tag="Falcon512_SKR"
)

_ = NTRU.estimate(Falcon512_SKR)
```

```
usvp                 :: rop: ≈2^165.1, red: ≈2^165.1, δ: 1.003489, β: 483, …
bdd                  :: rop: ≈2^160.6, red: ≈2^159.6, svp: ≈2^159.6, β: 463, …
bdd_hybrid           :: rop: ≈2^160.6, red: ≈2^159.6, svp: ≈2^159.6, β: 463, …
bdd_mitm_hybrid      :: rop: ≈2^349.3, red: ≈2^349.3, svp: ≈2^204.8, β: 481, …
```

H/T: Hunter Kippen added this!

Approaches

Unique SVP/BDD: Translation

We can reformulate $\mathbf{c} - \mathbf{A} \cdot \mathbf{s} \equiv \mathbf{e} \bmod q$ over the Integers as:

$$\begin{pmatrix} q\mathbf{I} & -\mathbf{A} \\ 0 & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} * \\ \mathbf{s} \end{pmatrix} + \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ \mathbf{s} \end{pmatrix}$$

Alternatively:

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I} & -\mathbf{A} & \mathbf{c} \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{B} \cdot \begin{pmatrix} * \\ \mathbf{s} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ \mathbf{s} \\ 1 \end{pmatrix}$$

In other words, there exists an integer-linear combination of the columns of $\mathbf{B}$ that produces a vector with “unusually” small entries $\rightarrow$ a unique shortest vector.

Unique SVP: Computational Problem

Unique Shortest Vector Problem for q-ary Lattices

Find a unique shortest vector amongst the integer combinations of the columns of:

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I} & -\mathbf{A} & \mathbf{c} \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where $\mathbf{B} \in \mathbb{Z}^{d \times d}$.

Decision Variant

Decide if $\mathbf{B}$ has an unusually short vector.

Approx SVP/SIS: Translation

- Consider $\mathbf{c} \equiv \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \bmod q$ with both $\mathbf{s}$ and $\mathbf{e}$ short or $\mathbf{c}$ uniform.
- Let $\mathbf{u}_i$ be short vectors such that $\mathbf{v}_i^T := \mathbf{u}_i^T \cdot \mathbf{A} \bmod q$ is also short.
- Compare:
  - $\mathbf{u}_i^T \cdot \mathbf{c} \equiv \mathbf{u}_i^T \cdot \mathbf{A} \cdot \mathbf{s} + \mathbf{u}_i^T \cdot \mathbf{e} \equiv \mathbf{v}_i^T \cdot \mathbf{s} + \mathbf{u}_i^T \cdot \mathbf{e}$ which is somewhat short
  - $\mathbf{u}_i^T \cdot \mathbf{c}$ which is uniform
- The shorter $(\mathbf{u}_i, \mathbf{v}_i)$ the fewer samples of $\mathbf{u}_i^T \cdot \mathbf{c}$ we need to consider
- Note

$$\begin{pmatrix} q\mathbf{I} & \mathbf{A}^T \\ 0 & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} * \\ \mathbf{u}_i \end{pmatrix} = \begin{pmatrix} \mathbf{v}_i \\ \mathbf{u}_i \end{pmatrix}$$

Approx SVP: Computational Problem

Short Vectors Problem for q-ary Lattices

Find vectors $(\mathbf{u}_i, \mathbf{v}_i)$ of norm $\|(\mathbf{u}_i, \mathbf{v}_i)\| \leq \beta$ amongst the integer combinations of the columns of:

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I} & \mathbf{A}^T \\ 0 & \mathbf{I} \end{pmatrix}$$

where $\mathbf{B} \in \mathbb{Z}^{d \times d}$.

Search Variant

Can extend this distinguishing attack to recover $\mathbf{s}$: guess a component and run the distinguisher

Lattice Reduction <3 Combinatorics: Hybrid Attacks

Both approaches can be augmented with a combinatorial step

- guess parts of the secret and run the lattice attack on a smaller dimensional lattice
- due to linearity costs are additive not multiplicative, i.e.

$$\approx T_{guess} + T_{lattice}$$

Estimator (Primal)

plain uSVP

```
LWE.primal_usvp(Kyber768)
```

```
rop: ≈2^204.9, red: ≈2^204.9, δ: 1.002902, β: 624, d: 1427, tag: usvp
```

plain BDD (minor parameter relaxation compared to uSVP)

```
LWE.primal_bdd(Kyber768)
```

```
rop: ≈2^201.0, red: ≈2^200.0, svp: ≈2^200.0, β: 606, η: 641, d: 1425, tag: bdd
```

BDD + combinatorics

```
LWE.primal_hybrid(Kyber768)
```

```
rop: ≈2^360.1, red: ≈2^359.4, svp: ≈2^358.8, β: 623, η: 2, ζ: 177, |S|: ≈2^366.3, d: 1290, prob: ≈2^-
152.8, ↻: ≈2^155.0, tag: hybrid
```

Estimator (Dual)

plain SIS

```
LWE.dual(Kyber768)
```

```
rop: ≈2^214.2, mem: ≈2^142.8, m: 723, β: 653, d: 1491, ↻: 1, tag: dual
```

SIS + combinatorics

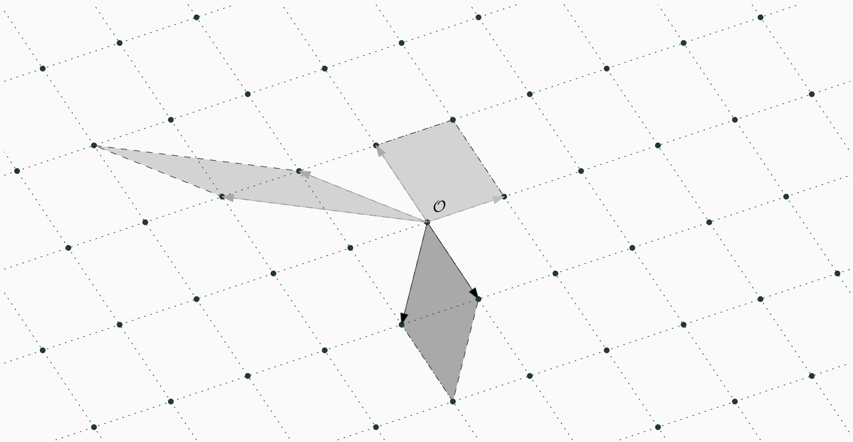
```
LWE.dual_hybrid(Kyber768)
```

```
rop: ≈2^196.1, red: ≈2^195.5, guess: ≈2^194.4, β: 586, p: 4, ζ: 15, t: 70, β': 580, N: ≈2^120.2, m: 768
```

Lattice Reduction

Lattice Volume

The volume of a lattice is the volume of its fundamental parallelepiped.



Picture Credit: Joop van de Pol

Gaussian Heuristic

- The Gaussian heuristic predicts that the number $|\Lambda \cap \mathcal{B}|$ of lattice points inside a measurable body $\mathcal{B} \subset \mathbb{R}^d$ is approximately equal to $\mathrm{Vol}(\mathcal{B}) / \mathrm{Vol}(\Lambda)$.
- Applied to Euclidean d-balls, this means that a shortest vector in a lattice has expected norm

$$\lambda_1(\Lambda) \approx \mathrm{GH}(d) \cdot \mathrm{Vol}(\Lambda)^{1/d} \approx \sqrt{\frac{d}{2\pi e}} \cdot \mathrm{Vol}(\Lambda)^{1/d}.$$

Unusually Shortest Vector

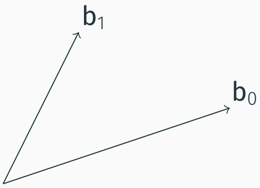
When $\lambda_1(\Lambda) \ll \sqrt{\frac{d}{2\pi e}} \cdot \mathrm{Vol}(\Lambda)^{1/d}$.

Length of Gram–Schmidt Vectors

It will be useful to consider the lengths of the Gram–Schmidt vectors.

The vector $\mathbf{b}_i^*$ is the orthogonal projection of $\mathbf{b}_i$ to the space spanned by the vectors $\mathbf{b}_0, \ldots, \mathbf{b}_{i-1}$.

Informally, this means taking out the contributions in the directions of previous vectors $\mathbf{b}_0, \ldots, \mathbf{b}_{i-1}$.



We have $\mathrm{Vol}(\Lambda) = \prod_{i=0}^{d-1} \|\mathbf{b}_i^*\|$.

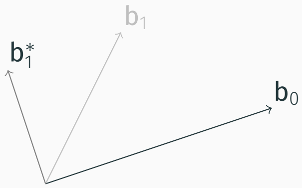
Length of Gram–Schmidt Vectors

It will be useful to consider the lengths of the Gram–Schmidt vectors.

The vector $\mathbf{b}_i^*$ is the orthogonal projection of $\mathbf{b}_i$ to the space spanned by the vectors $\mathbf{b}_0, \ldots, \mathbf{b}_{i-1}$.

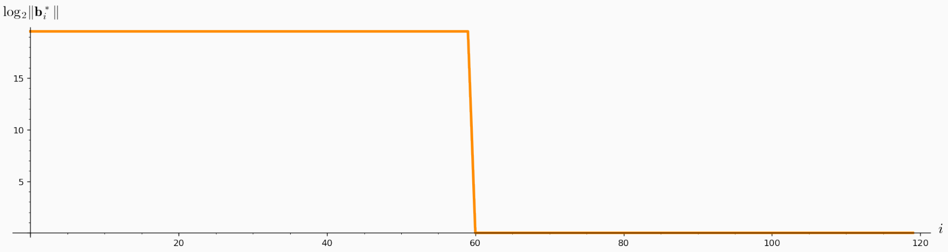
Informally, this means taking out the contributions in the directions of previous vectors $\mathbf{b}_0, \ldots, \mathbf{b}_{i-1}$.

We have $\mathrm{Vol}(\Lambda) = \prod_{i=0}^{d-1} \|\mathbf{b}_i^*\|$.



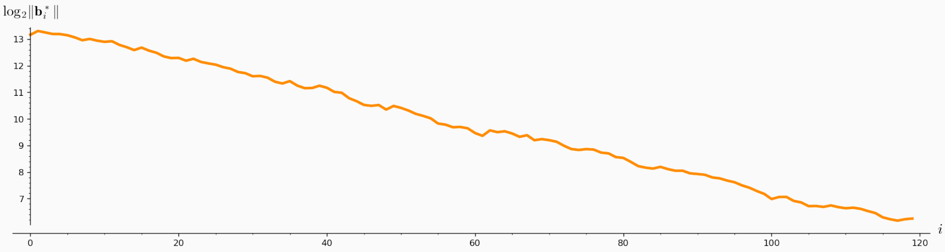
Example

```
A = IntegerMatrix.random(120, "qary", k=60, bits=20)[::-1]
M = GSO.Mat(A, update=True)
line([(i,log(r_, 2)/2) for i, r_ in enumerate(M.r())], **plot_kwds)
```

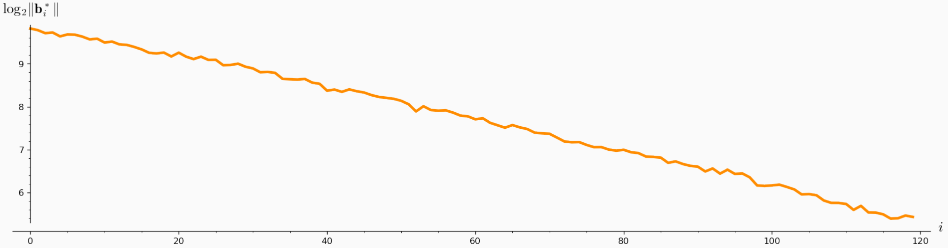


Example - LLL

```
A = LLL.reduction(A)
M = GSO.Mat(A, update=True)
line([(i,log(r_, 2)/2) for i, r_ in enumerate(M.r())], **plot_kwds)
```



GSA



Geometric Series Assumption: The shape after lattice reduction is a line with a flatter slope as lattice reduction gets stronger.[1]

[1] Claus-Peter Schnorr. Lattice Reduction by Random Sampling and Birthday Methods. In: *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*. Ed. by Helmut Alt and Michel Habib. Vol. 2607. Lecture Notes in Computer Science. Springer, 2003, pp. 145–156. DOI: `10.1007/3-540-36494-3_14`. URL: `http://dx.doi.org/10.1007/3-540-36494-3_14`.

Strong Lattice Reduction: BKZ Algorithm (Block 0)

$$\begin{pmatrix} \overbrace{\mid \quad\quad\quad\quad\quad\quad\quad\quad \mid}^{\beta = 5} & & & \\ b_0 \quad b_1 \quad b_2 \quad b_3 \quad b_4 & b_5 & b_6 & b_7 & \dots \\ \mid \quad\quad\quad\quad\quad\quad\quad\quad \mid & & & \end{pmatrix}$$

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm (Block 0)

$$\begin{pmatrix} \overbrace{\begin{matrix} \mathbf{b}_0 & \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 & \mathbf{b}_4 \end{matrix}}^{\beta = 5} & \mathbf{b}_5 & \mathbf{b}_6 & \mathbf{b}_7 & \dots \end{pmatrix}$$

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm (Block 0)

$$\overbrace{}^{\beta = 5}$$

$$\begin{pmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & \ldots \end{pmatrix}$$

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm (Block 1)

$$\begin{pmatrix} \mathbf{b}_0 & \overbrace{\pi_0(\mathbf{b}_1) \quad \pi_0(\mathbf{b}_2) \quad \pi_0(\mathbf{b}_3) \quad \pi_0(\mathbf{b}_4) \quad \pi_0(\mathbf{b}_5)}^{\beta = 5} & \mathbf{b}_6 & \mathbf{b}_7 & \ldots \end{pmatrix}$$

$\pi_i(\mathbf{v})$: project $\mathbf{v}$ orthogonally to $\mathbf{b}_0, \ldots, \mathbf{b}_i$

Strong Lattice Reduction: BKZ Algorithm (Block 1)

$$\left(\mathbf{b}_0 \quad \underbrace{\overbrace{\pi_0(\mathbf{b}_1) \quad \pi_0(\mathbf{b}_2) \quad \pi_0(\mathbf{b}_3) \quad \pi_0(\mathbf{b}_4) \quad \pi_0(\mathbf{b}_5)}^{\beta = 5}} \quad \mathbf{b}_6 \quad \mathbf{b}_7 \quad \dots \right)$$

$\pi_i(\mathbf{v})$: project $\mathbf{v}$ orthogonally to $\mathbf{b}_0, \dots, \mathbf{b}_i$

Strong Lattice Reduction: BKZ Algorithm (Block 1)

$$\begin{pmatrix} \mathbf{b}_0 & \overbrace{\pi_0(\mathbf{b}_1) & \pi_0(\mathbf{b}_2) & \pi_0(\mathbf{b}_3) & \pi_0(\mathbf{b}_4) & \pi_0(\mathbf{b}_5)}^{\beta = 5} & \mathbf{b}_6 & \mathbf{b}_7 & \ldots \end{pmatrix}$$

$\pi_i(\mathbf{v})$: project $\mathbf{v}$ orthogonally to $\mathbf{b}_0, \ldots, \mathbf{b}_i$

Data: LLL-reduced lattice basis **B**
Data: block size β
repeat *until no more change*
  for $\kappa \leftarrow 0$ **to** $d - 1$ **do**
    LLL on local projected block $[\kappa, \ldots, \kappa + \beta - 1]$;
    v $\leftarrow$ find shortest vector in local projected block $[\kappa, \ldots, \kappa + \beta - 1]$;
    insert **v** into **B**;
  end

Quality

For SIS
$\|\mathbf{b}_0\| \approx {\delta_\beta}^{d-1} \cdot \mathrm{Vol}(\Lambda)^{1/d}$

For BDD
$\|\mathbf{b}_0\| \approx {\delta_\beta}^{2\cdot(d-\beta)} \cdot \lambda_1(\Lambda)$

β	2	5	24	50	100	200	500
δ_β	1.0219	1.0186	1.0142	1.0121	1.0096	1.0063	1.0034

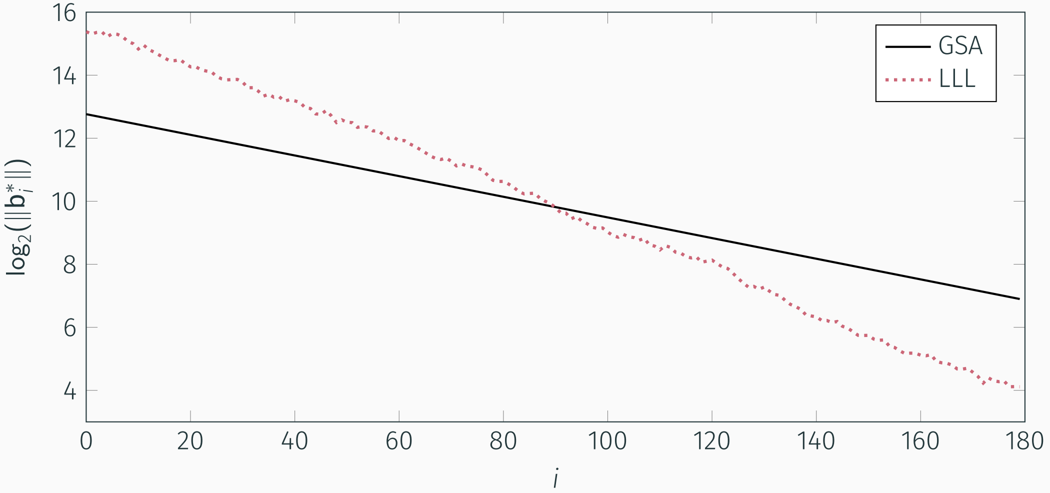
- We have **Root Hermite Factor** $\delta_\beta \approx \mathrm{GH}(\beta)^{1/(\beta-1)}$ for $\beta > 50$.

```
RC.delta(500)
```

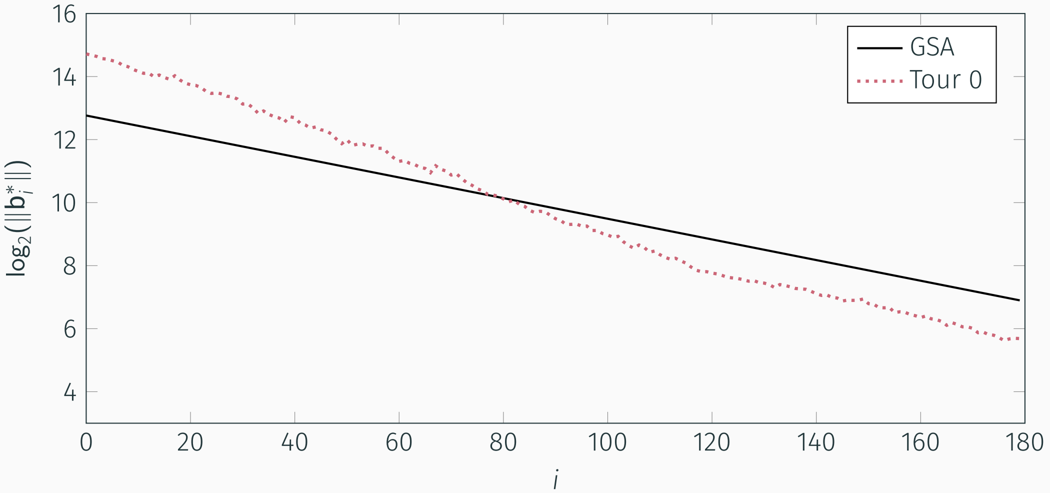
```
1.00340402678510
```

- The slope under the **Geometric Series Assumption** is $\alpha_\beta = \delta_\beta^{-2}$.

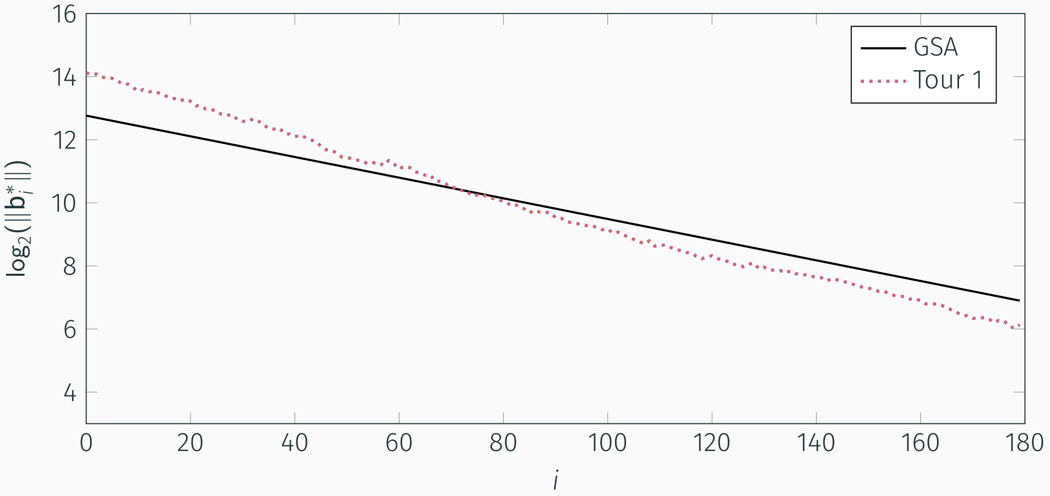
Behaviour in Practice: BKZ-60 in Dimension 180 i



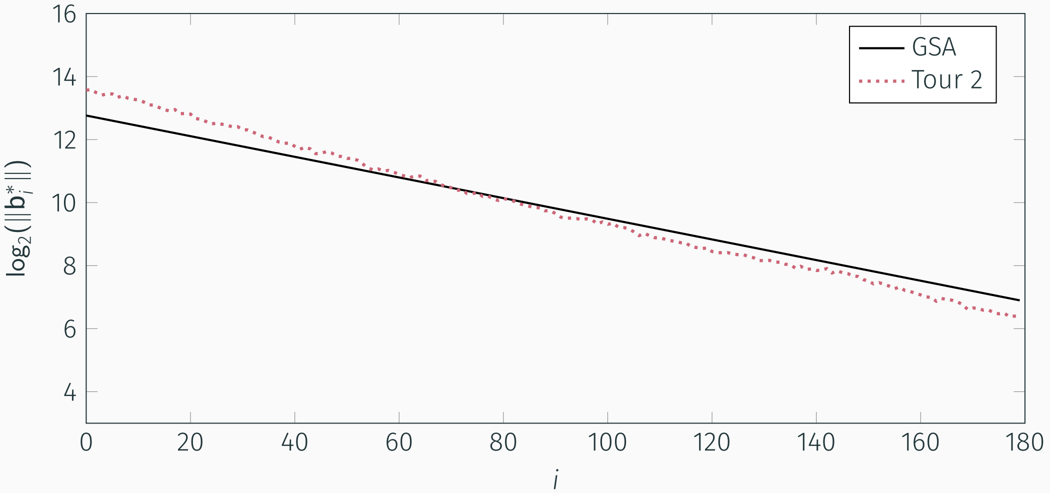
Behaviour in Practice: BKZ-60 in Dimension 180 II



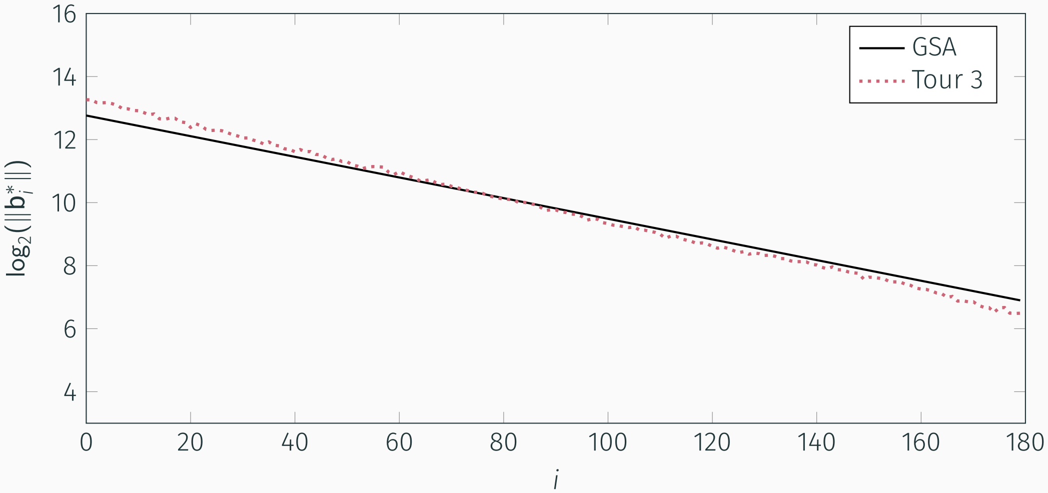
Behaviour in Practice: BKZ-60 in Dimension 180 III



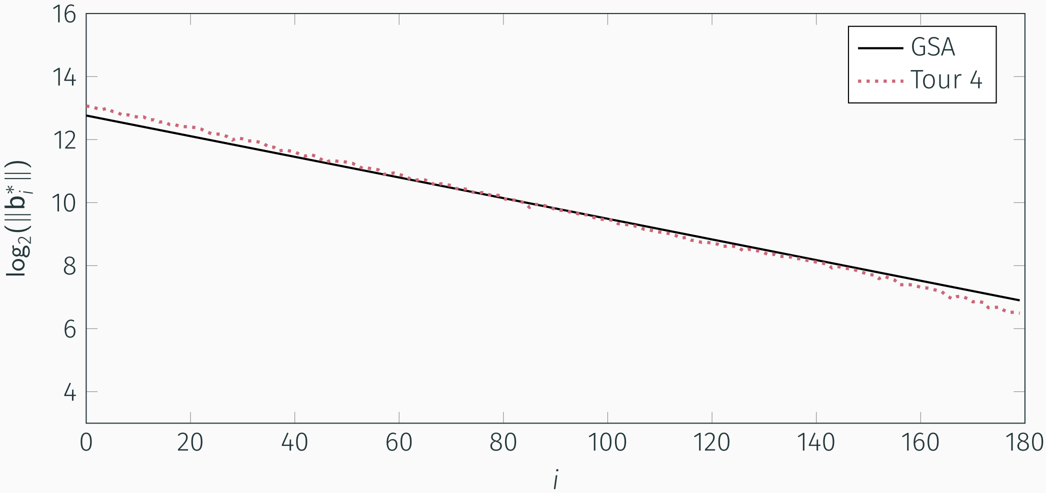
Behaviour in Practice: BKZ-60 in Dimension 180 IV



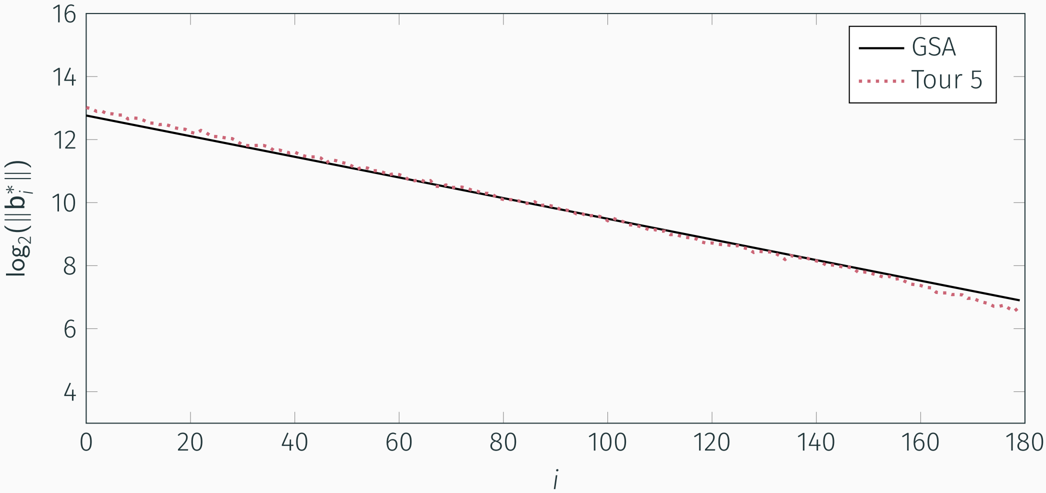
Behaviour in Practice: BKZ-60 in Dimension 180 V



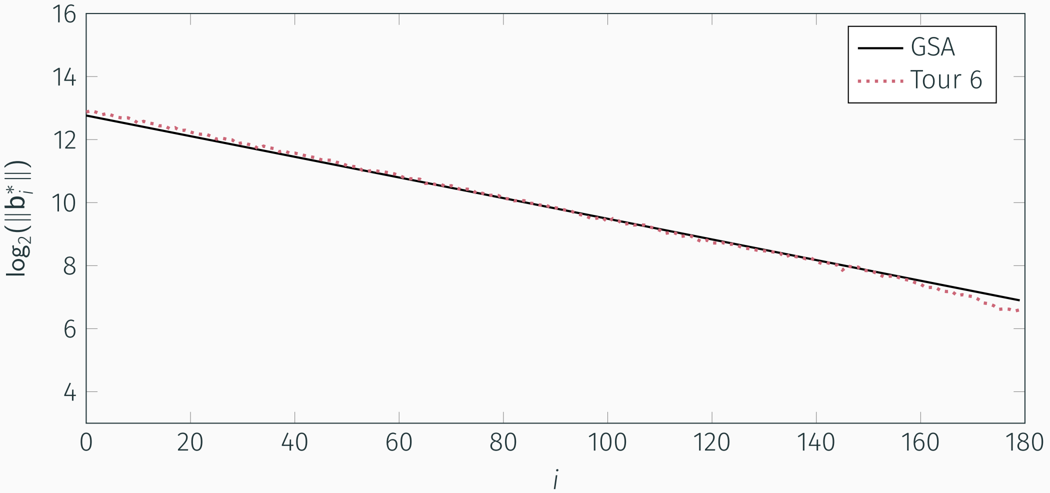
Behaviour in Practice: BKZ-60 in Dimension 180 vi



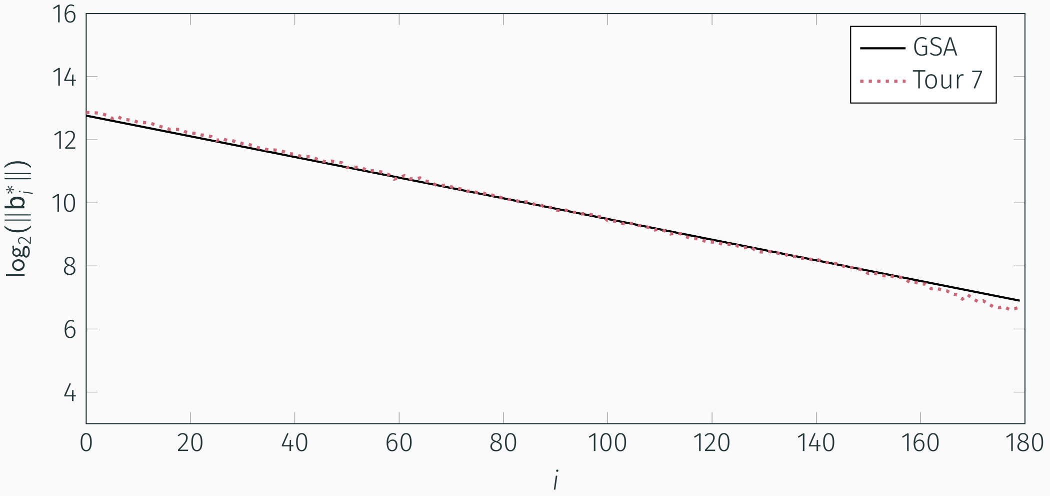
Behaviour in Practice: BKZ-60 in Dimension 180 vii



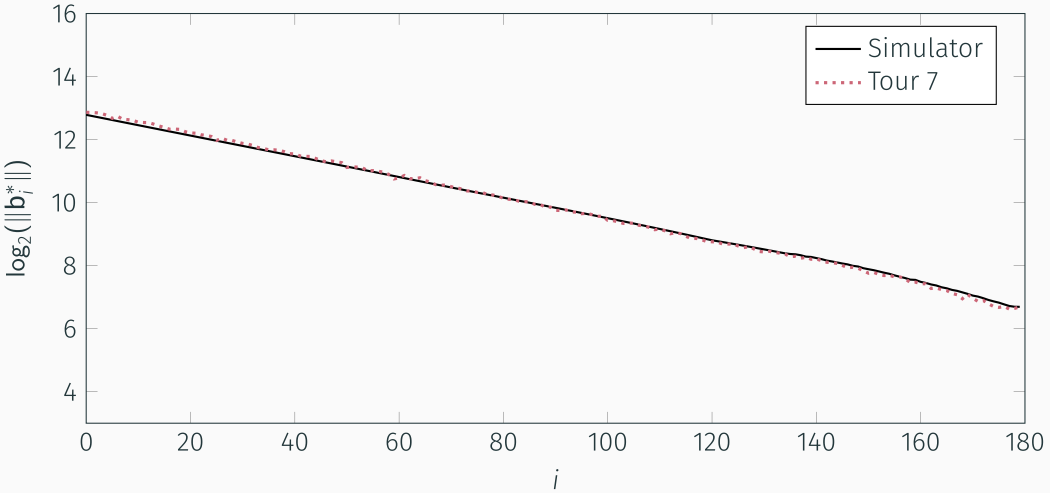
Behaviour in Practice: BKZ-60 in Dimension 180 viii



Behaviour in Practice: BKZ-60 in Dimension 180 ix



Behaviour in Practice: BKZ-60 in Dimension 180 x



Try it at Home

```python
from fpylll import *
from fpylll.algorithms.bkz2 import BKZReduction as BKZ2
A = IntegerMatrix.random(180, "qary", k=90, bits=20)
bkz = BKZ2(A)
bkz(BKZ.EasyParam(block_size=60))
```

`https://github.com/fplll/fplll` C++ library

`https://github.com/fplll/fpylll` Python interface

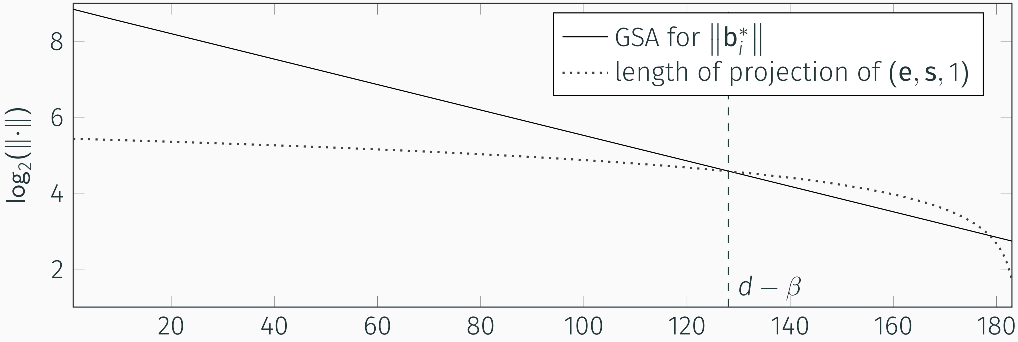
`https://github.com/fplll/g6k` Sieving (faster lattice reduction)

`https://sagemath.org` FPyLLL is in SageMath

`https://sagecell.sagemath.org/` SageMath in your browser

`https://cocalc.com/` SageMath worksheets in your browser

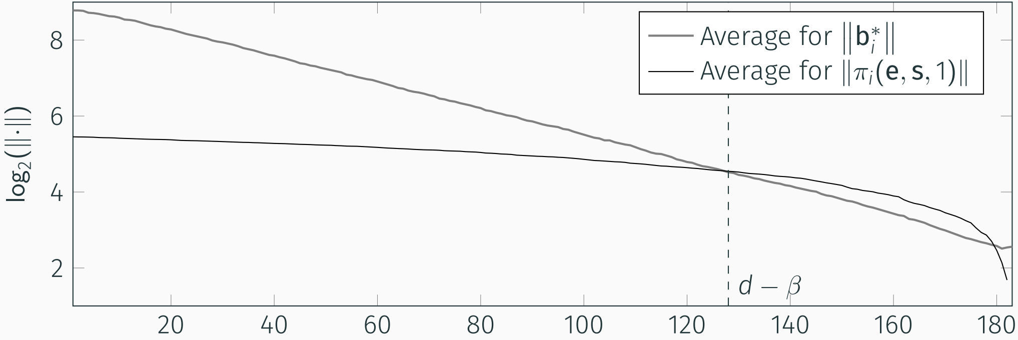
Success Condition for uSVP (Expectation)



$$\sqrt{\beta/d} \cdot \sigma \leq \delta_\beta{}^{2\beta-d-1} \cdot \mathrm{Vol}(\Lambda)^{1/d}$$

Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum Key Exchange - A New Hope. In: *USENIX Security 2016*. Ed. by Thorsten Holz and Stefan Savage. USENIX Association, Aug. 2016, pp. 327–343

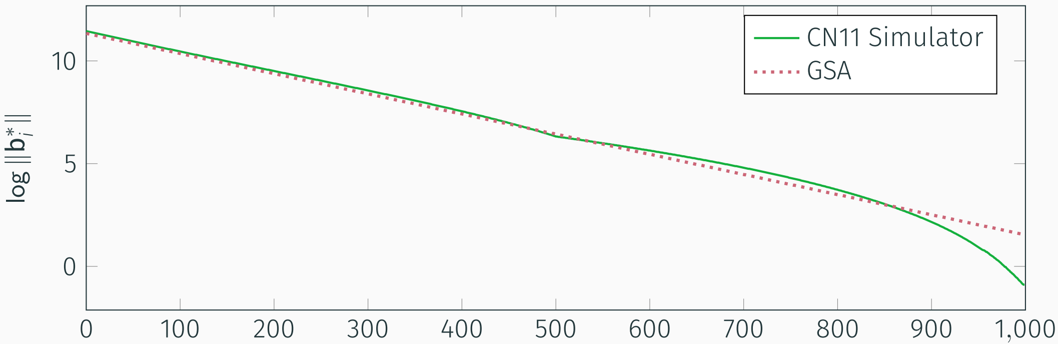
Success Condition for uSVP (Observed)



Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the Expected Cost of Solving uSVP and Applications to LWE. In: *ASIACRYPT 2017, Part I*. ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. LNCS. Springer, Heidelberg, Dec. 2017, pp. 297–322. DOI: `10.1007/978-3-319-70694-8_11`

Eamonn W. Postlethwaite and Fernando Virdia. On the Success Probability of Solving Unique SVP via BKZ. In: *PKC 2021, Part I*. ed. by Juan Garay. Vol. 12710. LNCS. Springer, Heidelberg, May 2021, pp. 68–98. DOI: `10.1007/978-3-030-75245-3_4`

The GSA is a Lie: Tail Shape



Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In: *ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, Heidelberg, Dec. 2011, pp. 1–20. DOI: 10.1007/978-3-642-25385-0_1

The GSA is a Lie: Tail Shape

```python
from estimator import *
print(repr(LWE.primal_usvp(Kyber768, red_shape_model="GSA")))  # used in LWE.estimate.rough
print(repr(LWE.primal_usvp(Kyber768, red_shape_model="CN11"))) # used in LWE.estimate
```

```
rop: ≈2^204.9, red: ≈2^204.9, δ: 1.002902, β: 624, d: 1427, tag: usvp
rop: ≈2^209.9, red: ≈2^209.9, δ: 1.002842, β: 642, d: 1421, tag: usvp
```

Cost

- If τ is the number of tours we do, we run our oracle $\approx \tau \cdot d$ times
- So the cost is roughly $\tau \cdot d \cdot T_{SVP}$.
- We can reduce some of this cost

  Tail is cheaper than the head as we decrease the block sizes

  Progressive BKZ Run BKZ-β' with $\beta' < \beta$ before running BKZ-β

  Skipping blocks We may get away with "skipping" some blocks.
- `LWE.estimate.rough` assumes **one** call to the oracle ("Core-SVP")
- `LWE.estimate` assumes roughly $8 \cdot d$, i.e. $\tau = 8$

```
print(repr(LWE.primal_usvp(Kyber768, red_cost_model=RC.ADPS16))) # used in LWE.estimate.rough
print(repr(LWE.primal_usvp(Kyber768, red_cost_model=RC.MATZOV))) # used in LWE.estimate
```

```
rop: ≈2^182.2, red: ≈2^182.2, δ: 1.002902, β: 624, d: 1427, tag: usvp
rop: ≈2^204.9, red: ≈2^204.9, δ: 1.002902, β: 624, d: 1427, tag: usvp
```

Reading Estimator Output

```
LWE.primal_bdd(Kyber768, red_shape_model="CN11")
```

```
rop: ≈2^204.0, red: ≈2^203.1, svp: ≈2^202.8, β: 617, η: 651, d: 1457, tag: bdd
```

rop elementary operations ("ring operations" for some reason)

red elementary operations during lattice reduction

β BKZ block size

η dimension of final oracle call

d lattice dimension

Ad Break: We now do SIS, too!

```
Dilithium2_MSIS_WkUnf = SIS.Parameters(
    n=256*4,
    q=8380417,
    length_bound=350209,
    m=256*9,
    norm=oo,
    tag="Dilithium2_MSIS_WkUnf"
)
_ = SIS.estimate(Dilithium2_MSIS_WkUnf)
```

```
lattice  :: rop: ≈2^152.2, red: ≈2^151.3, sieve: ≈2^151.1, β: 427, η: 433, …
```

H/T: Hunter Kippen added this!

Solving SVP

Solving SVP

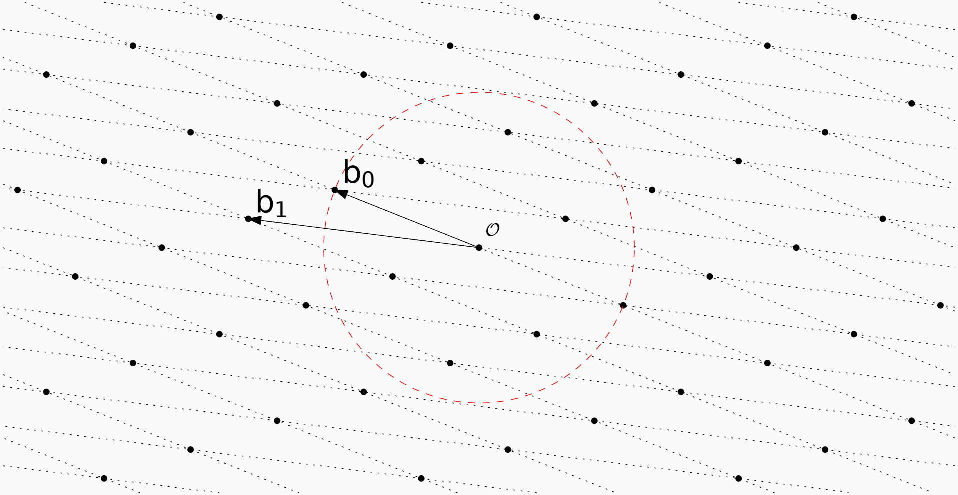
Enumeration

- Search through vectors smaller than a given bound: project down to 1-dim problem, lift to 2-dim problem …
- Sensitive to the quality of the input basis
- **Time:** $2^{\Theta(\beta \log \beta)}$
- **Memory:** $\mathrm{poly}(\beta)$

Sieving

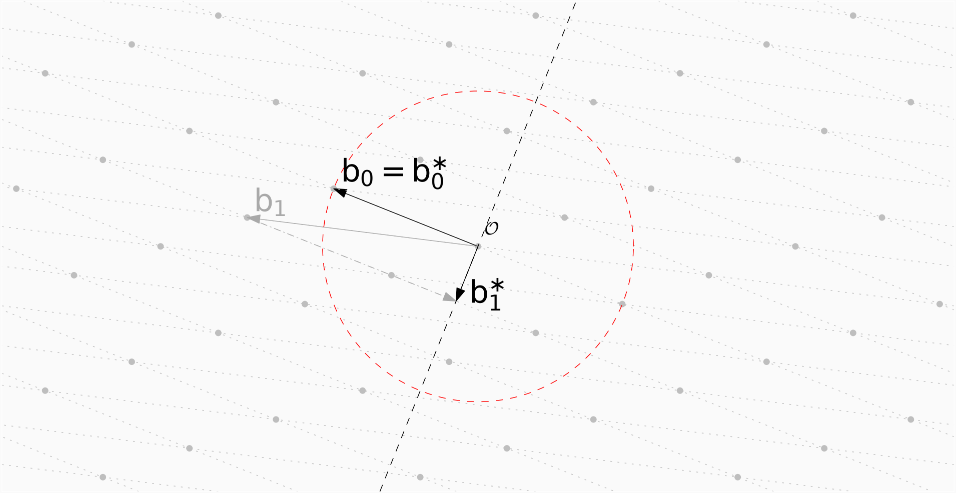
- Produce new, shorter vectors by considering sums and differences of existing vectors
- Fairly oblivious to the quality of the input basis
- **Time:** $2^{\Theta(\beta)}$
- **Memory:** $2^{\Theta(\beta)}$

Enumeration I – Pick a radius



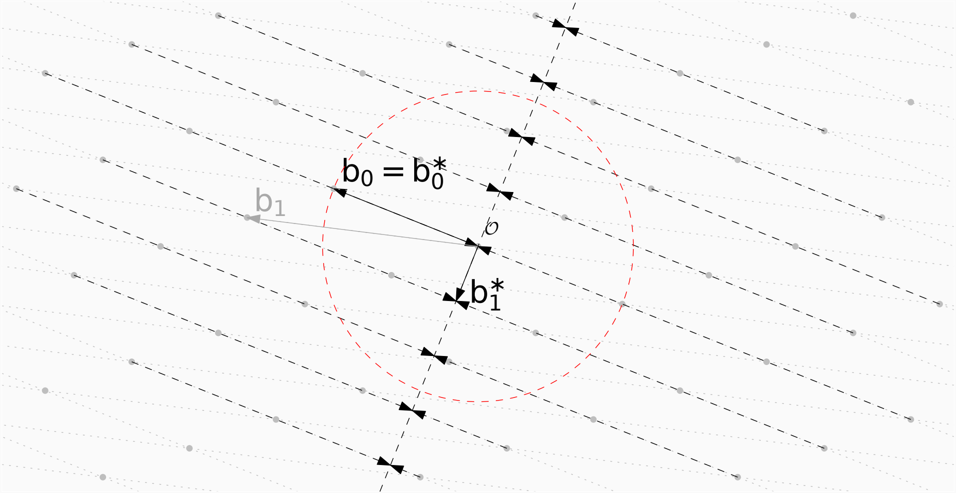
Picture credit: Joop van de Pol

Enumeration II – Project basis



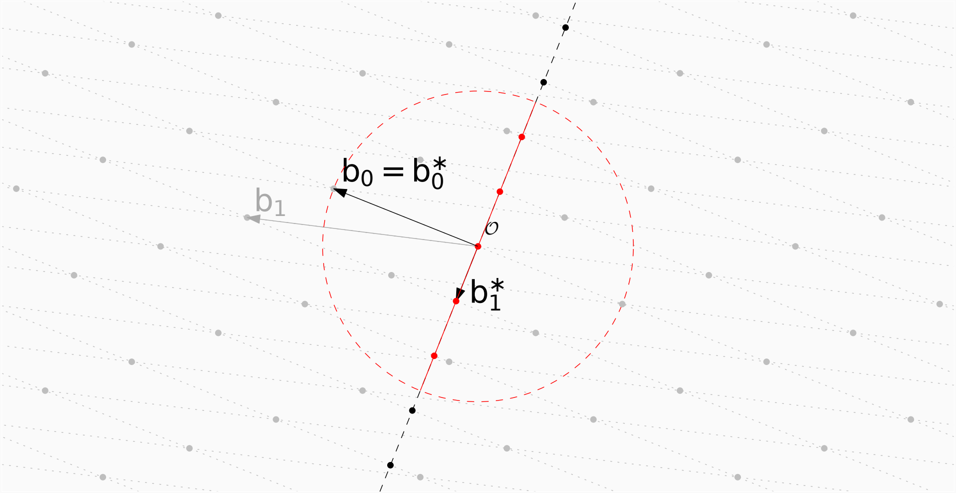
Picture credit: Joop van de Pol

Enumeration III – Project lattice



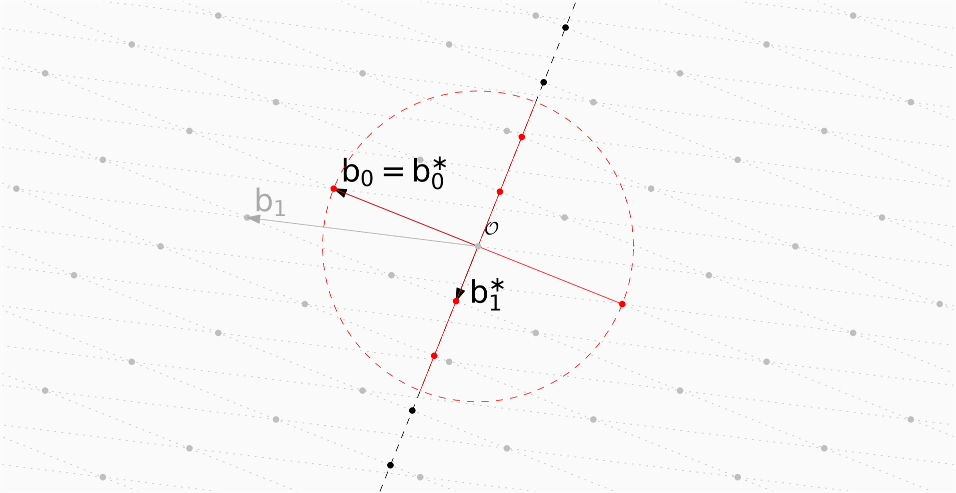
Picture credit: Joop van de Pol

Enumeration IV – Enumerate projections



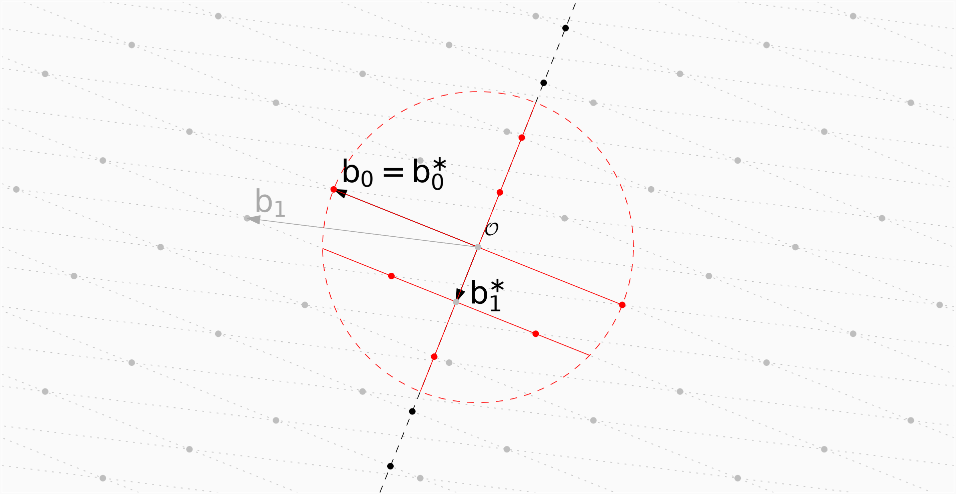
Picture credit: Joop van de Pol

Enumeration V – For each lift and enumerate



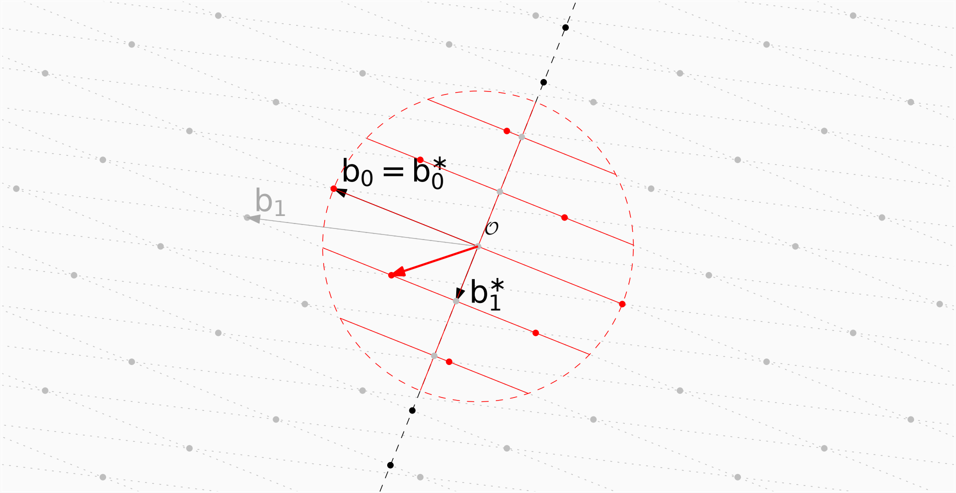
Picture credit: Joop van de Pol

Enumeration V – For each lift and enumerate



Picture credit: Joop van de Pol

Enumeration VI – Keep shortest

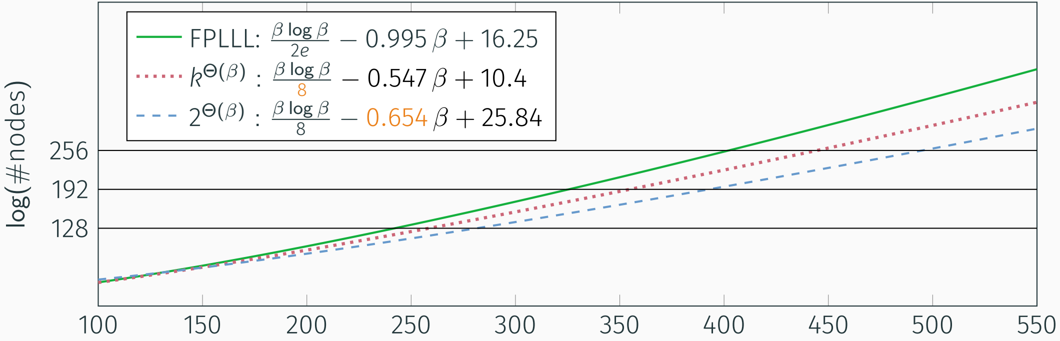


Picture credit: Joop van de Pol

Fast Enumeration

- Do not exhaust the search space, but focus on a fraction with exponentially small probability of success, repeat exponentially often: speed-up $2^{\Theta(\beta)}$
- Preprocess the basis with BKZ-β' for some $\beta' \leq \beta$ before enumerating.

Practical Performance (Simulation)

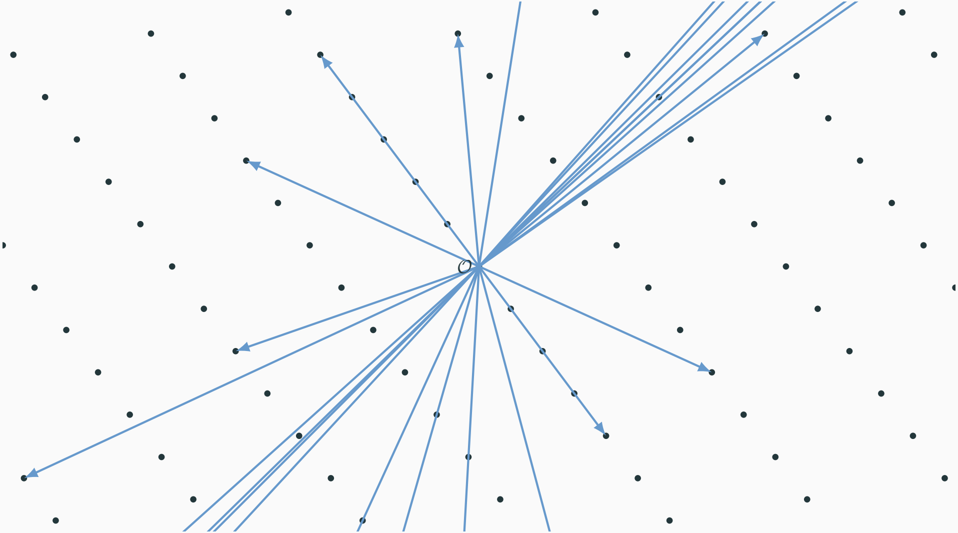



```
beta, d = 500, 1000
RC.CheNgu12(beta, d).log(2), RC.ABFKSW20(beta, d).log(2), RC.ABLR21(beta, d).log(2)
```

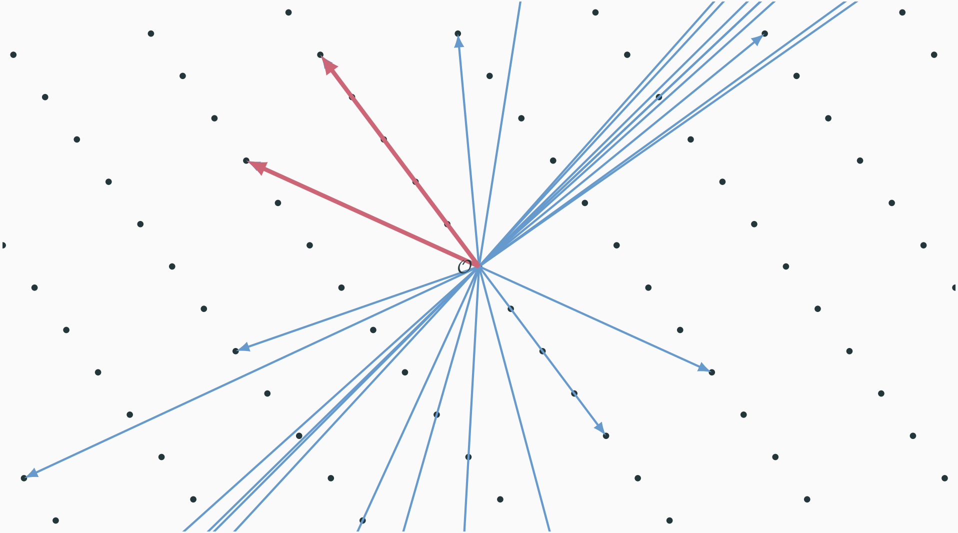
```
(365.668328064860, 316.227302076042, 278.167302076042)
```

[CN11; ABFKSW20; ABLR21]

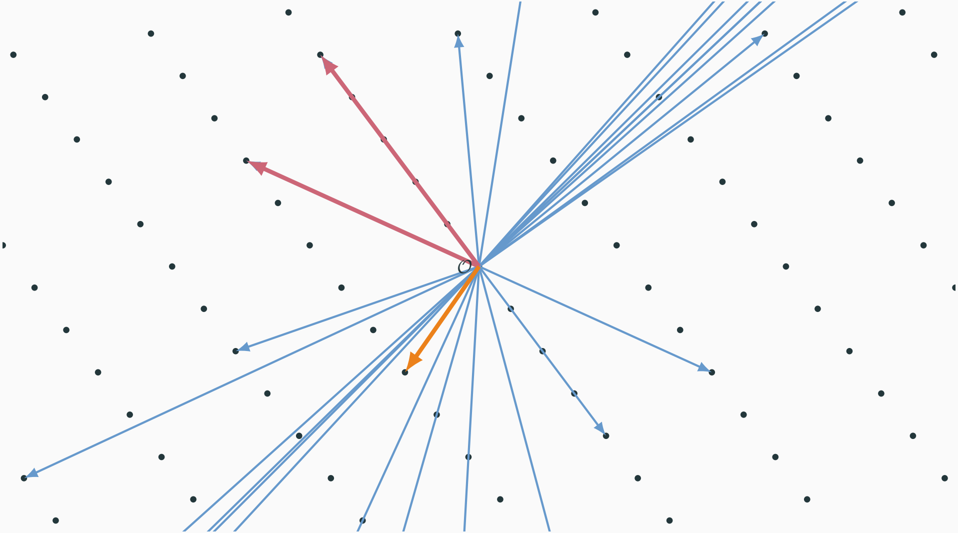
Sieving: Key Idea I



Sieving: Key Idea II



Sieving: Key Idea III



Sieving: Basic (Gauss) Sieve Complexity

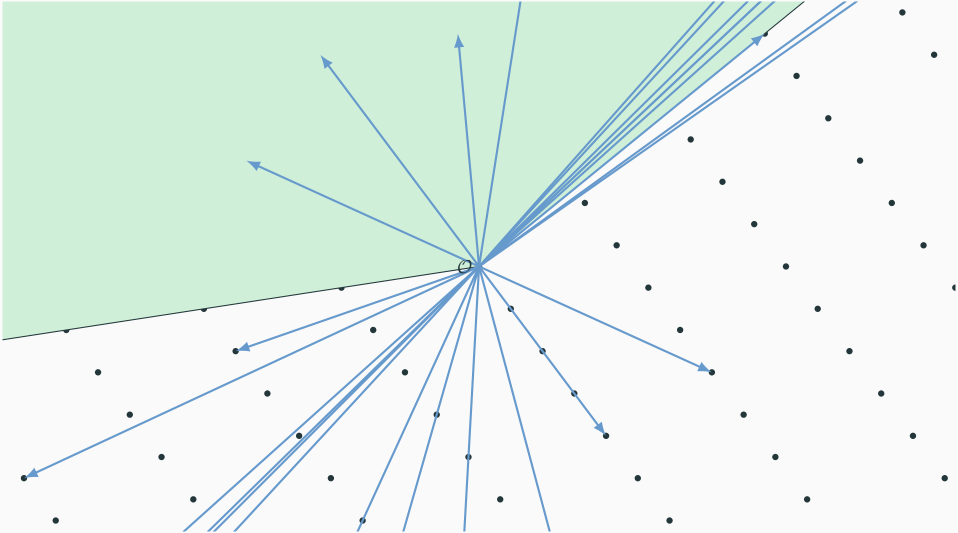
- Assume all vectors have (roughly) the same length
- Normalise to unit sphere $\mathcal{S}^{d-1} := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| = 1\}$
- We have $\|\mathbf{v} - \mathbf{w}\| \leq 1$ iff $\langle \mathbf{v}, \mathbf{w} \rangle \geq 1/2 = \cos(\pi/3)$
- The probability that two random $\mathbf{v}, \mathbf{w} \in \mathcal{S}^{d-1}$ satisfy $\langle \mathbf{v}, \mathbf{w} \rangle \geq 1/2$ is

$$= \mathrm{poly}(d) \cdot \left(\frac{4}{3}\right)^{d/2} \approx 2^{0.2075\, d + o(d)}$$

- Need $\mathrm{poly}(d) \cdot \left(\frac{4}{3}\right)^{d/2}$ vectors, comparing all pairs costs $\mathrm{poly}(d) \cdot \left(\frac{4}{3}\right)^{d} \approx 2^{0.4150\, d + o(d)}$.

Daniele Micciancio and Panagiotis Voulgaris. Faster Exponential Time Algorithms for the Shortest Vector Problem. In: *21st SODA*. ed. by Moses Charika. ACM-SIAM, Jan. 2010, pp. 1468–1480. DOI: 10.1137/1.9781611973075.119

Sieving: Buckets I



Sieving: Buckets II

If **v**, **c** are somewhat close and **w**, **c** are somewhat close then perhaps **w**, **v** are close?

Strategy

- Sort vectors into somewhat loose buckets,
- Do quadratic pairwise comparison only within each bucket.

BGJ Split search space into buckets. **Cost**: $2^{0.311\,\beta+o(\beta)}$.[2]

BDGL Use codes to decide which bucket to consider. **Cost**: $2^{0.292\,\beta+o(\beta)}$.[3]

---

[2] Anja Becker, Nicolas Gama, and Antoine Joux. Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search. Cryptology ePrint Archive, Report 2015/522. `https://eprint.iacr.org/2015/522`. 2015.

[3] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In: *27th SODA*. ed. by Robert Krauthgamer. ACM-SIAM, Jan. 2016, pp. 10–24. DOI: `10.1137/1.9781611974331.ch2`.

Sieving: G6K

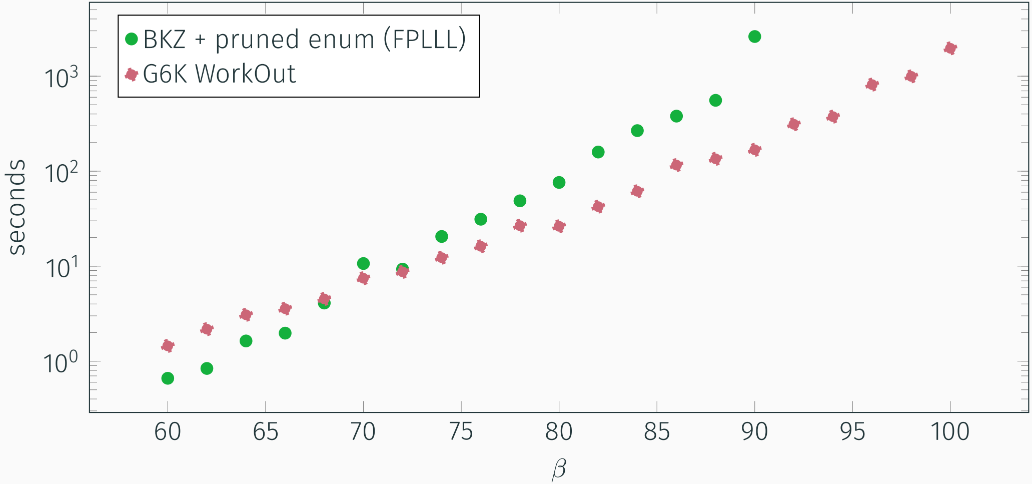
G6K[4] is a Python/C++ framework for experimenting with sieving algorithms (inside and outside BKZ)

- Does not take the "oracle" view but considers sieves as stateful machines.
- Implements several sieve algorithms
  - Gauss and NV
  - Triple Sieve
  - BGJ1 (BGJ with one level of filtration)
  - BDGL (with one and two block respectively)
- Applies recent tricks and adds new tricks for improving performance of sieving

---

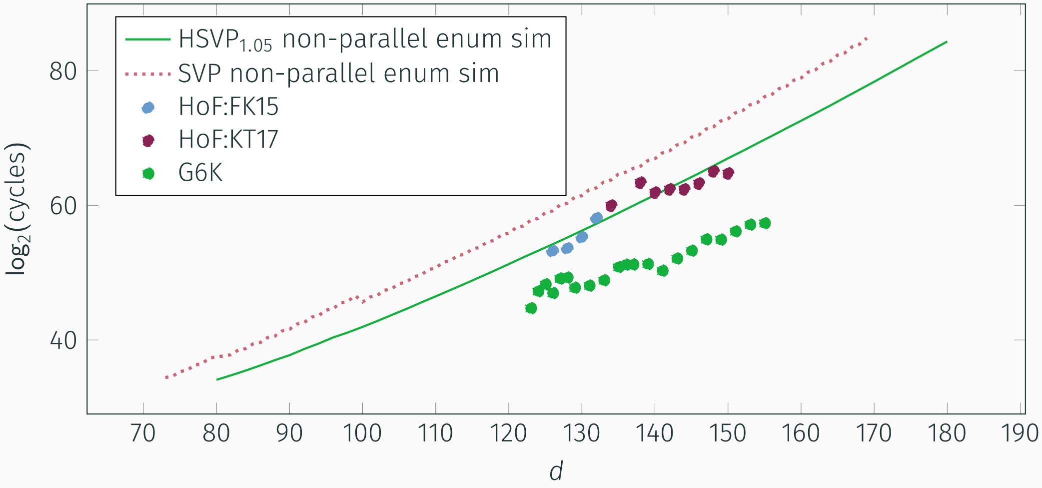
[4] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The General Sieve Kernel and New Records in Lattice Reduction. In: *EUROCRYPT 2019, Part II*. ed. by Yuval Ishai and Vincent Rijmen. Vol. 11477. LNCS. Springer, Heidelberg, May 2019, pp. 717–746. DOI: `10.1007/978-3-030-17656-3_25`.

Sieving: SVP



Average time in seconds for solving exact SVP

Darmstadt $HSVP_{1.05}$ Challenges



GPU Sieving

- Stream database of vectors to GPU
- Run low precision inner products there

dim	TD4F	D4F	MSD	Norm	Norm/GH	FLOP	Walltime	Mem GiB
158	31	29	129	3303	1.04329	262.1	9h 16m	89
162	31	31	131	3341	1.04220	263.2	18h 32m	156
176	34	33	143	3487	1.04412	267.5	12d 11h	806
178	34	32	146	3447	1.02725	268.6	22d 18h	1060
180	34	30	150	3509	1.04003	269.9	51d 14h	1443

EC:DucSteWoe21

Try it at Home

```python
from fpylll import IntegerMatrix, GSO, LLL
from fpylll.tools.bkz_stats import dummy_tracer
from g6k import Siever
from g6k.algorithms.bkz import pump_n_jump_bkz_tour

A = LLL.reduction(IntegerMatrix.random(180, "qary", k=90, bits=20))
g6k = Siever(A)

for b in range(20, 60+1, 10):
    pump_n_jump_bkz_tour(g6k, dummy_tracer, b, pump_params={"down_sieve": True})
```

`https://github.com/fplll/g6k` C++ kernel + Python frontend

`https://github.com/WvanWoerden/G6K-GPU-Tensor` G6K fork adding GPU support

Costing Sieves

"The main difference is the cost of the random product code decoding algorithm."

MATZOV. Report on the Security of LWE: Improved Dual Lattice Attack. Available at `https://doi.org/10.5281/zenodo.6412487`. Apr. 2022. DOI: `10.5281/zenodo.6412487`. URL: `https://doi.org/10.5281/zenodo.6412487`

"Concretely, we conclude on an overhead factor of about on the number of gates in the RAM model compared to the idealized model for dimensions around after an appropriate re-parametrization."

Léo Ducas. Estimating the Hidden Overheads in the BDGL Lattice Sieving Algorithm. Cryptology ePrint Archive, Report 2022/922. `https://eprint.iacr.org/2022/922`. 2022

"Core-SVP" [ADPS16]: $2^{0.292\,\beta\pm 0}$ v [Sch+20; AGPS20] v [MAT22]

```
RC.ADPS16(500, 1000).log(2), RC.Kyber(500, 1000).log(2), RC.MATZOV(500, 1000).log(2)
```

```
(146.000000000000, 176.547704482770, 169.704298365530)
```

ML Attacks

A Series of Recent Works

- Emily Wenger, Mingjie Chen, François Charton, and Kristin E. Lauter. **SALSA: Attacking Lattice Cryptography with Transformers**. In: *Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022.* Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh. 2022
- Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Mohamed Malhou, Evrard Garcelon, François Charton, and Kristin E. Lauter. **SalsaPicante: A Machine Learning Attack on LWE with Binary Secrets**. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023.* Ed. by Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda. ACM, 2023, pp. 2606–2620
- Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Zeyuan Allen-Zhu, François Charton, and Kristin E. Lauter. **SALSA VERDE: a machine learning attack on Learning with Errors with sparse small secrets**. 2023. URL: `https://eprint.iacr.org/2023/968`
- Samuel Stevens, Emily Wenger, Cathy Yuanchen Li, Niklas Nolte, Eshika Saxena, François Charton, and Kristin E. Lauter. **SALSA FRESCA: Angular Embeddings and Pre-Training for ML Attacks on Learning With Errors**. 2024. URL: `https://eprint.iacr.org/2024/150`

Claims I

Ethics and Broader Impact. *The primary value of this work is in alerting the cryptographic and ML communities to the risk of ML-based attacks on PQC. Even if current attacks do not succeed, we believe that* ***providing early warning of potential threats is critical****. However, we emphasize that SALSA represents a proof of concept that cannot be used against real-world implementations (i.e. the PQC schemes which NIST standardized on July 5, 2022). Additional scaling work would be necessary before these techniques would be relevant to attacking real-world cryptosystems.” – [WCCL22]*

Claims II

Ethical considerations. *Although Picante demonstrates significant progress towards attacking real-world LWE problems with sparse binary secrets, **it cannot yet break** problems with real-world-size parameters. In particular, the LWE schemes standardized by NIST use smaller modulus q and non-sparse secret distributions. Hence, we do not believe our paper raises any ethical concerns. Nonetheless, we shared a copy of the current paper with the NIST Cryptography group, to inform them of our approach.* – [LSWMGCL23]

Claims III

Limitations and broader impact. *Despite significantly advancing the state-of-the-art in ML-based LWE attacks, VERDE* ***cannot yet break*** *standardized LWE-based PQC schemes, limiting its real-world impact. Because of this, our paper raises no immediate security concerns. Nevertheless, we have shared a copy of our paper with the NIST PQC group to make them aware of this attack.* – [LSWACL23]

CLAIMS IV

*8. **Impact Statement** The main ethical concern related to this work is the possibility of our attack compromising currently-deployed PQC system. However, **at present, our proposed attack does not threaten current standardized systems**. If our attack scales to higher h and lower q settings, then its impact is significant, as it would necessitate changing PQC encryption standards. For reproducability of these results, our code will be open sourced after publication and is available to reviewers upon request. – [SWLNSCL24]*

Attack Description

The preprocessing step strives to reduce the norm of the rows of $\mathbf{A}$ by applying a carefully selected integer linear operator $\mathbf{R}$. Because $\mathbf{R}$ is linear with integer entries, the transformed pairs $(\mathbf{RA}, \mathbf{Rb}) \mod \mathbf{q}$ are also LWE pairs with the same secret, albeit larger error. In practice, $\mathbf{R}$ is found by performing lattice reduction on the $(m+n) \times (m+n)$ matrix $\mathbf{\Lambda} = \begin{bmatrix} 0 & q \cdot \mathbf{I}_n \\ \omega \cdot \mathbf{I}_m & \mathbf{A} \end{bmatrix}$, and finding linear operators $\begin{bmatrix} \mathbf{C} & \mathbf{R} \end{bmatrix}$ such that the norms of $\begin{bmatrix} \mathbf{C} & \mathbf{R} \end{bmatrix} \mathbf{\Lambda} = \begin{bmatrix} \omega \cdot \mathbf{R} & \mathbf{RA} + q \cdot \mathbf{C} \end{bmatrix}$ are small. This achieves a reduction of the norms of the entries of $\mathbf{RA} \mod q$, but also increases the error in the calculation of $\mathbf{Rb} = \mathbf{RA} \cdot \mathrm{s} + \mathbf{Re}$, making secret recovery more difficult. Although ML models can learn from noisy data, too much noise will make the distribution of $\mathbf{Rb}$ uniform on $[0, q)$ and inhibit learning. The parameter ω controls the trade-off between norm reduction and error increase. Reduction strength is measured by $\rho = \frac{\sigma(\mathbf{RA})}{\sigma(\mathbf{A})}$, where σ denotes the mean of the standard deviations of the rows of $\mathbf{RA}$ and $\mathbf{A}$.

Li et al. (2023a) use BKZ (Schnorr, 1987) for lattice reduction. Li et al. (2023b) improves the reduction time by $45\times$ via a modified definition of the Λ matrix and by interleaving `BKZ2.0` (Chen & Nguyen, 2011) and `polish` (Charton et al., 2024) (see Appendix C).

This preprocessing step produces many $(\mathbf{RA}, \mathbf{Rb})$ pairs that can be used to train models. Individual rows of $\mathbf{RA}$ and associated elements of $\mathbf{Rb}$, denoted as reduced LWE samples $(\mathbf{Ra}, \mathbf{R}b)$ with some abuse of notation, are used for model training. Both the subsampling of m samples from the original t LWE samples and the reduction step are

Recent versions of the attack (VERDE/FRESCA) are essentially variants of the dual attack.

- $\mathbf{u}^T \cdot \mathbf{c} \equiv \mathbf{u}^T \cdot \mathbf{A} \cdot \mathbf{s} + \mathbf{u}^T \cdot \mathbf{e} \equiv \mathbf{v}^T \cdot \mathbf{s} + \mathbf{u}^T \cdot \mathbf{e} \Rightarrow$ short-ish
- $\mathbf{u}^T \cdot \mathbf{c} \Rightarrow$ uniform

Distinguishers

Modelling $\mathbf{v}^T \cdot \mathbf{s} + \mathbf{u}^T \cdot \mathbf{e}$ as a discrete Gaussian mod q we can compute the statistical distance between these two distributions and thus the number of samples we need to distinguish with constant advantage.

Comparison with State of the Art: SALSA VERDE I

Table 15. Comparison of VERDE's and uSVP attack performance on LWE problems with $n = 256$, binary secrets, varying q and h. *VERDE's total attack time is the sum of preprocessing and training time (with recovery included). Preprocessing time assumes full parallelization, and training time is the number of epochs to recovery multiplied by epoch time (1.5 hours/epoch). N/A means no successful secret recovery.*

LWE parameters		**VERDE attack time**			**uSVP attack time (hrs)**
$\log_2 q$	h	*Preprocessing (hrs)*	*Training*	*Total (hrs)*	
12	8	1.5	2 epochs	4.5	N/A
14	12	2.5	2-5 epochs	5.5-10	N/A
16	14	8.0	2 epochs	11	N/A
18	18	7.0	3 epochs	11.5	558
18	20	7.0	1-8 epochs	8.5-19	259
20	22	7.5	5 epochs	15	135-459
20	23	7.5	3-4 epochs	12-15	167-330
20	24	7.5	4 epochs	13.5	567
20	25	7.5	5 epochs	15	76 - 401

To summarize the comparison, VERDE outperforms existing classical attacks in two senses: 1) VERDE fully recovers sparse binary and ternary secrets for n and q where existing classical attacks do not succeed in several weeks or months using *fplll* BKZ 2.0 [19] with the required block size;

https://crypto.iacr.org/2023/rump/crypto2023rump-paper13.pdf

Comparison with State of the Art: SALSA VERDE II

Table 15. Comparison of VERDE's and uSVP attack performance on LWE problems with $n = 256$, binary secrets, varying q and h. *VERDE's total attack time is the sum of preprocessing and training time (with recovery included). Preprocessing time assumes full parallelization, and training time is the number of epochs to recovery multiplied by epoch time (1.5 hours/epoch). N/A means no successful secret recovery.*

LWE parameters		VERDE attack time			
$\log_2 q$	h	*Preprocessing (hrs)*	*Training*	*Total*	(hrs × CPU)
12	8	1.5	2 epochs	4.5	× ???
14	12	2.5	2-5 epochs	5.5-10	× 270
16	14	8.0	2 epochs	11	× ???
18	18	7.0	3 epochs	11.5	× 990
18	20	7.0	1-8 epochs	8.5-19	× ???
20	22	7.5	5 epochs	15	× ???
20	23	7.5	3-4 epochs	12-15	× ???
20	24	7.5	4 epochs	13.5	× ???
20	25	7.5	5 epochs	15	× ???

State of the Art Attack (hrs, 1core)

0.2 **(MITM ... in Py)**

Implementation in Progress

(Hybrid MITM-Lattice) Models and predictions, exists, but no open source implem.

12 -- 24
(rescaled uSVP)

To summarize the comparison, VERDE is several orders of magnitude behind the state of the art, even on these custom made instances.

https://crypto.iacr.org/2023/rump/crypto2023rump-paper13.pdf

Comparison with Something: SALSA FESCA I

SALSA FRESCA: Angular Embeddings and Pre-Training for ML Attacks on LWE

Table 1. **Best results from our attack** for LWE problems in dimensions n (higher is harder), modulus q (lower is harder) and Hamming weights h (higher is harder). Our work recovers secrets for $n = 1024$ for the first time in ML-based LWE attacks and reduces total attack time for $n = 512, \log_2 q = 41$ to only 50 hours (assuming full CPU parallelization).

n	$\mathbf{\log_2 q}$	**highest h**	**LWE (A, b) matrices needed**	**preprocessing time** (hrs/CPU/matrix)	**training time** (hrs)	**total time** (hrs)
512	41	44	1955	13.1	36.9	50.0
768	35	9	1302	12.4	14.8	27.2
1024	50	13	977	26.0	47.4	73.4

```
from estimator import *
params = LWE.Parameters(n=1024, q=2^50, Xs=ND.SparseTernary(n=1024, p=7, m=7), Xe=ND.DiscreteGaussian(3))
LWE.primal_hybrid(params)
```

```
rop: ≈2^48.4, red: ≈2^48.1, svp: ≈2^46.2, β: 41, η: 2, ζ: 478, |S|: ≈2^42.6, d: 1213, prob: 0.189, ↻: 22, …
```

$$\approx 52 \text{ hrs vs } 977 \cdot 26 + 47.4 \approx 25402 \text{ hrs}$$

Comparison with Something: SALSA FESCA II

The Lattice Estimator picks $\beta = 40$ as a lower bound, it is not designed to handle such easy instances.

```
with local_minimum(40, max(2 * params.n, 41), precision=5) as it:
    for beta in it:
        cost = self.cost_gsa(
            beta=beta, params=params, m=m, red_cost_model=red_cost_model, **kwds
        )
        it.update(cost)
    for beta in it.neighborhood:
        cost = self.cost_gsa(
            beta=beta, params=params, m=m, red_cost_model=red_cost_model, **kwds
        )
        it.update(cost)
    cost = it.y
```

https://github.com/malb/lattice-estimator/blob/main/estimator/lwe_primal.py#L209-L220

High-Level

There is no particular reason to believe that ML can threaten LWE.

On Lattices, Learning with Errors,
Random Linear Codes, and Cryptography

Oded Regev *

May 2, 2009

Abstract

Our main result is a reduction from worst-case lattice problems such as GapSVP and SIVP to a certain learning problem. This learning problem is a natural extension of the 'learning from parity with error' problem to higher moduli. It can also be viewed as the problem of decoding from a random linear code. This, we believe, gives a strong indication that these problems are hard. Our reduction, however, is quantum. Hence, an efficient solution to the learning problem implies a *quantum* algorithm for GapSVP and SIVP. A main open question is whether this reduction can be made classical (i.e., non-quantum).

We also present a (classical) public-key cryptosystem whose security is based on the hardness of the learning problem. By the main result, its security is also based on the worst-case quantum hardness of GapSVP and SIVP. The new cryptosystem is much more efficient than previous lattice-based cryptosystems: the public key is of size $\tilde{O}(n^2)$ and encrypting a message increases its size by a factor of $\tilde{O}(n)$ (in previous cryptosystems these values are $\tilde{O}(n^4)$ and $\tilde{O}(n^2)$, respectively). In fact, under the assumption that all parties share a random bit string of length $\tilde{O}(n^2)$, the size of the public key can be reduced to $\tilde{O}(n)$.

- **LWE** is (designed to be) a hard learning problem.
- **ML** classifiers exploit statistical patterns in the data.[a]

Open Problem

Not easy to establish the state of the art for LWE instances within range of experiments. More advanced algorithms lack efficient, versatile and public implementations.

[a] This is a reason why they work somewhat well on e.g. side-channel traces.

Thank You

KCL Academic staff, postdocs and PhD students (all areas of cryptography)

SandboxAQ Postdoc/PhD/FTEs/Consultants: PQC PhD residencies, PQC Postdocs, Cryptography SWE

[ABFKSW20] Martin R. Albrecht, Shi Bai, Pierre-Alain Fouque, Paul Kirchner, Damien Stehlé, and Weiqiang Wen. Faster Enumeration-Based Lattice Reduction: Root Hermite Factor $k^{1/(2k)}$ Time $k^{k/8+o(k)}$. In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020, pp. 186–212. DOI: 10.1007/978-3-030-56880-1_7.

[ABLR21] Martin R. Albrecht, Shi Bai, Jianwei Li, and Joe Rowell. Lattice Reduction with Approximate Enumeration Oracles - Practical Algorithms and Concrete Performance. In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 732–759. DOI: 10.1007/978-3-030-84245-1_25.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Heidelberg, Aug. 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35.

[ADHKPS19] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The General Sieve Kernel and New Records in Lattice Reduction. In: *EUROCRYPT 2019, Part II*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11477. LNCS. Springer, Heidelberg, May 2019, pp. 717–746. DOI: 10.1007/978-3-030-17656-3_25.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum Key Exchange - A New Hope. In: *USENIX Security 2016*. Ed. by Thorsten Holz and Stefan Savage. USENIX Association, Aug. 2016, pp. 327–343.

References II

[AGPS20] Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck. Estimating Quantum Speedups for Lattice Sieves. In: *ASIACRYPT 2020, Part II*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12492. LNCS. Springer, Heidelberg, Dec. 2020, pp. 583–613. DOI: `10.1007/978-3-030-64834-3_20`.

[AGVW17] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the Expected Cost of Solving uSVP and Applications to LWE. In: *ASIACRYPT 2017, Part I*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. LNCS. Springer, Heidelberg, Dec. 2017, pp. 297–322. DOI: `10.1007/978-3-319-70694-8_11`.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In: *27th SODA*. Ed. by Robert Krauthgamer. ACM-SIAM, Jan. 2016, pp. 10–24. DOI: `10.1137/1.9781611974331.ch2`.

[BGJ15] Anja Becker, Nicolas Gama, and Antoine Joux. Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search. Cryptology ePrint Archive, Report 2015/522. `https://eprint.iacr.org/2015/522`. 2015.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In: *ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, Heidelberg, Dec. 2011, pp. 1–20. DOI: `10.1007/978-3-642-25385-0_1`.

[Duc22] Léo Ducas. Estimating the Hidden Overheads in the BDGL Lattice Sieving Algorithm. Cryptology ePrint Archive, Report 2022/922. `https://eprint.iacr.org/2022/922`. 2022.

[LSWACL23] Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Zeyuan Allen-Zhu, François Charton, and Kristin E. Lauter. SALSA VERDE: a machine learning attack on Learning with Errors with sparse small secrets. 2023. URL: https://eprint.iacr.org/2023/968.

[LSWMGCL23] Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Mohamed Malhou, Evrard Garcelon, François Charton, and Kristin E. Lauter. SalsaPicante: A Machine Learning Attack on LWE with Binary Secrets. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023*. Ed. by Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda. ACM, 2023, pp. 2606–2620.

[MAT22] MATZOV. Report on the Security of LWE: Improved Dual Lattice Attack. Available at https://doi.org/10.5281/zenodo.6412487. Apr. 2022. DOI: 10.5281/zenodo.6412487. URL: https://doi.org/10.5281/zenodo.6412487.

[MV10] Daniele Micciancio and Panagiotis Voulgaris. Faster Exponential Time Algorithms for the Shortest Vector Problem. In: *21st SODA*. Ed. by Moses Charika. ACM-SIAM, Jan. 2010, pp. 1468–1480. DOI: 10.1137/1.9781611973075.119.

[PV21] Eamonn W. Postlethwaite and Fernando Virdia. On the Success Probability of Solving Unique SVP via BKZ. In: *PKC 2021, Part I*. Ed. by Juan Garay. Vol. 12710. LNCS. Springer, Heidelberg, May 2021, pp. 68–98. DOI: 10.1007/978-3-030-75245-3_4.

References IV

[Sch03] Claus-Peter Schnorr. Lattice Reduction by Random Sampling and Birthday Methods. In: *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*. Ed. by Helmut Alt and Michel Habib. Vol. 2607. Lecture Notes in Computer Science. Springer, 2003, pp. 145–156. DOI: `10.1007/3-540-36494-3_14`. URL: `http://dx.doi.org/10.1007/3-540-36494-3_14`.

[Sch+20] Peter Schwabe et al. CRYSTALS-KYBER. Tech. rep. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`. National Institute of Standards and Technology, 2020.

[SWLNSCL24] Samuel Stevens, Emily Wenger, Cathy Yuanchen Li, Niklas Nolte, Eshika Saxena, François Charton, and Kristin E. Lauter. SALSA FRESCA: Angular Embeddings and Pre-Training for ML Attacks on Learning With Errors. 2024. URL: `https://eprint.iacr.org/2024/150`.

[WCCL22] Emily Wenger, Mingjie Chen, François Charton, and Kristin E. Lauter. SALSA: Attacking Lattice Cryptography with Transformers. In: *Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh. 2022.