So how hard is solving LWE/NTRU anyway?

Martin R. Albrecht @martinralbrecht

10 January 2019, RWC

Based on joint work with Alex Davidson, Amit Deo, Benjamin R. Curtis, Eamonn W. Postlethwaite, Elena Kirshanova, Fernando Virdia, Florian Göpfert, Gottfried Herold, Léo Ducas, Marc Stevens, Rachel Player, Sam Scott and Thomas Wunderer as well as the work of many other authors.

Introduction

NIST Process: Selected Non-Quantum Security Estimates

Scheme / Cost Model	Kyber	Lima	R EMBLEM	NTRU HRSS	SNTRU'
Kyber[1]	180	218	112	136	155
Lima[2]	196	234	129	152	171
R EMBLEM[3]	210	248	142	165	184
NTRU HRSS[4]	456	587	242	313	370
SNTRU'[5]	535	722	270	350	410

Source: Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate All the LWE, NTRU Schemes! In: *SCN 18*. Ed. by Dario Catalano and Roberto De Prisco. Vol. 11035. LNCS. Springer, Heidelberg, Sept. 2018, pp. 351–367. DOI: `10.1007/978-3-319-98113-0_19`, `https://estimate-all-the-lwe-ntru-schemes.github.io/docs/`

[1] 0.292β [Alk+16], this is an explicit underestimate
[2] $0.292\beta + 16.4$ [Sma+17], this is a somewhat explicit underestimate
[3] $0.292\beta + \log(8d) + 16.4$ [APS15]
[4] $0.18728\,\beta\log(\beta) - 1.0192\,\beta + 16.10 + 7$ [APS15]
[5] $0.000784314\,\beta^2 + 0.366078\,\beta - 6.125\log(8d) + 7$ [Hof+15]

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

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

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 coefficients.

Primal Attack

Unique SVP Approach

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 coefficients $\rightarrow$ a unique shortest vector.

Computational Problem

Unique Shortest Vector Problem

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

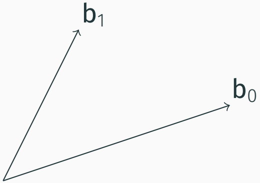
Lattice Reduction

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 $\mathbf{b}_0, \ldots, \mathbf{b}_{i-1}$.

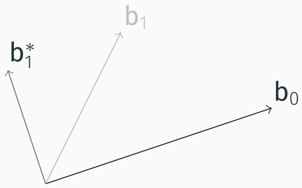


Length of Gram-Schmidt Vectors

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

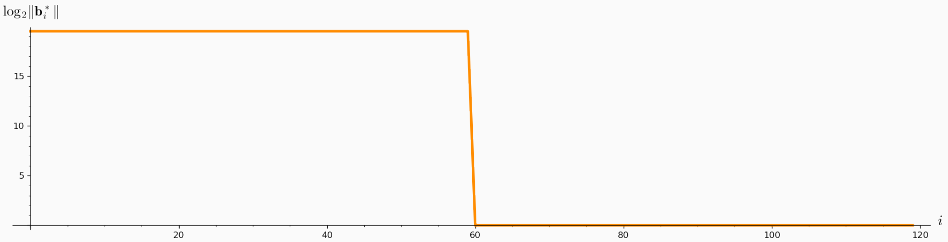
The vector $\mathbf{b}_i^*$ is the orthogonal projection of $\mathbf{b}_i$ to the space spanned by the vectors $\mathbf{b}_0, \ldots, \mathbf{b}_{i-1}$.

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



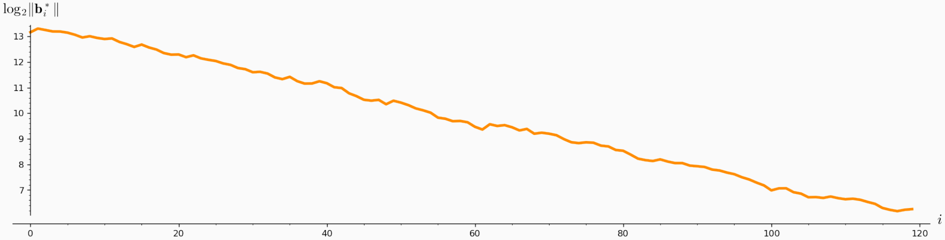
Example

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



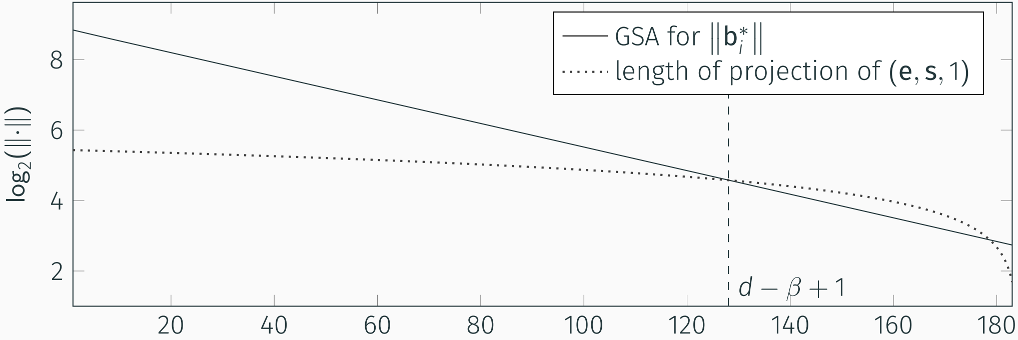
Example - LLL

```
sage: A = LLL.reduction(A)
sage: M = GSO.Mat(A); M.update_gso()
sage: lg = [(i,log(r_, 2)/2) for i, r_ in enumerate(M.r())]
sage: line(lg, **plot_kwds)
```



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

Success Condition for uSVP

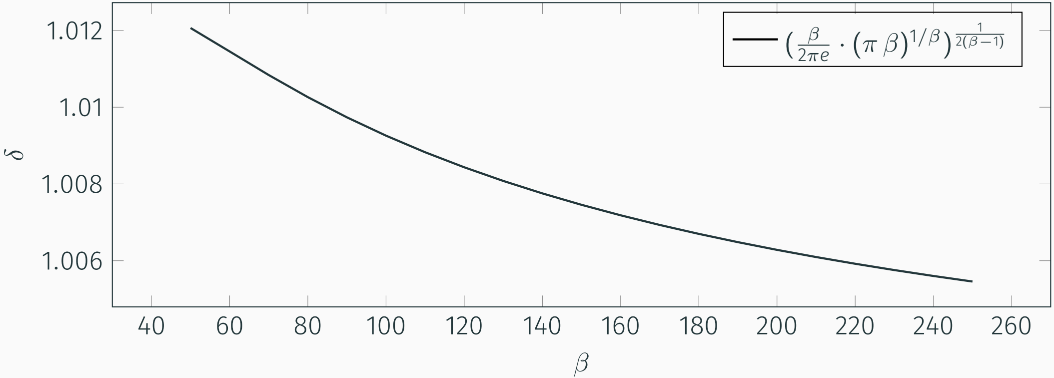


Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum Key Exchange - A New Hope. In: *25th USENIX Security Symposium, USENIX Security 16.* Ed. by Thorsten Holz and Stefan Savage. USENIX Association, 2016, pp. 327–343. URL: `https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim`

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`

Slope

The slope depends on the **root Hermite factor** δ which depends on the “block size” β.



Yuanmi Chen. Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. PhD thesis. Paris 7, 2013

Strong Lattice Reduction: BKZ Algorithm

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

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm

$$\left(\overbrace{b_0 \quad b_1 \quad b_2 \quad b_3 \quad b_4}^{\beta = 5} \quad b_5 \quad b_6 \quad b_7 \quad \dots \right)$$

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm

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

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm

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

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm

$$\left(\; b_0 \quad \overbrace{b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5}^{\beta = 5} \quad b_6 \quad b_7 \quad \dots \; \right)$$

Picture credit: Eamonn Postlethwaite

$$\left(\begin{array}{ccccccccc} b_0 & \underbrace{\begin{array}{|ccccc|} b_1 & b_2 & b_3 & b_4 & b_5 \end{array}}_{} & b_6 & b_7 & \dots \end{array} \right)$$

$\beta = 5$

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm

$$\left(\begin{array}{ccccccccc} \color{red}{b_0} & \color{red}{b_1} & \underbrace{b_2 & b_3 & b_4 & b_5 & b_6}_{} & b_7 & \ldots \end{array} \right)$$

$$\beta = 5$$

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm

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

Picture credit: Eamonn Postlethwaite

Strong Lattice Reduction: BKZ Algorithm

$$\left(\; b_0 \quad b_1 \quad \overbrace{b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6}^{\beta = 5} \quad b_7 \quad \ldots \; \right)$$

Picture credit: Eamonn Postlethwaite

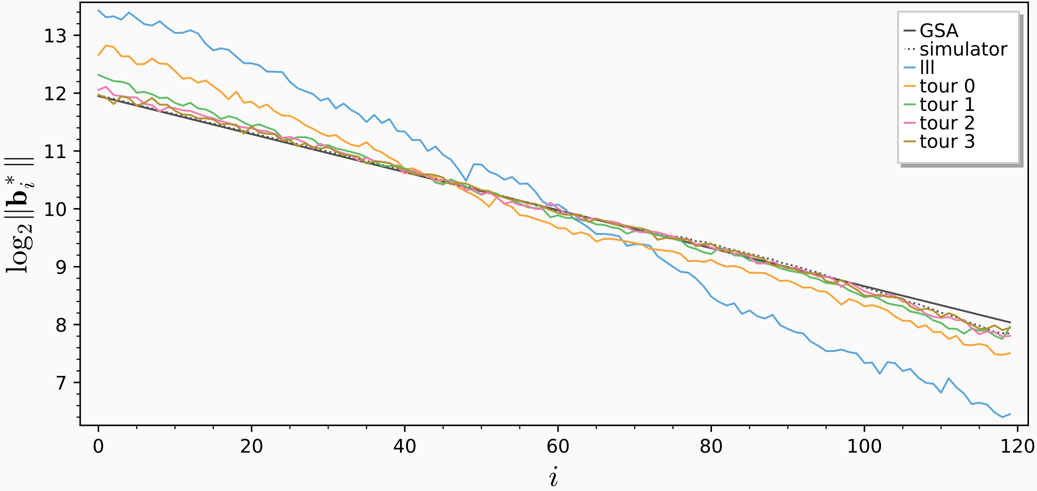
BKZ Algorithm

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 $[\kappa, \ldots, \kappa+\beta-1]$;
    $\mathbf{v} \leftarrow$ find shortest vector in local projected block $[\kappa, \ldots, \kappa+\beta-1]$;
    insert $\mathbf{v}$ into **B**;
  end

Jargon

An outer loop iteration is called a “tour”.

Behaviour in Practice: BKZ-60 in Dimension 120



Number of Tours

Scheme / Cost Model	Kyber	Lima	R EMBLEM	NTRU HRSS	SNTRU’
0.292β	180	218	112	136	155
$0.292\beta + 16.4$	196	234	129	152	171
$0.292\beta + \log(8d) + 16.4$	210	248	142	165	184
$0.18728\,\beta\log(\beta) - 1.0192\,\beta + 16.10 + 7$	456	587	242	313	370
$0.000784314\,\beta^2 + 0.366078\,\beta - 6.125 + \log(8d) + 7$	535	722	270	350	410

After 4 to 8 tours the output does not change much. Thus, some authors write $8d \cdot t_{SVP}$. Others argue that we need to call the SVP oracle at least once and write t_{SVP}.

Open Question

$8d$ is too large[6] but it is not clear how far this factor can be reduced in practice.

[6] Mingjie Liu and Phong Q. Nguyen. Solving BDD by Enumeration: An Update. In: *CT-RSA 2013*. Ed. by Ed Dawson. Vol. 7779. LNCS. Springer, Heidelberg, 2013, pp. 293–309. DOI: `10.1007/978-3-642-36095-4_19`.

Solving SVP

Solving SVP

Scheme / Cost Model	Kyber	Lima	R EMBLEM	NTRU HRSS	SNTRU'
0.292β	180	218	112	136	155
$0.292\beta + 16.4$	196	234	129	152	171
$0.292\beta + \log(8d) + 16.4$	210	248	142	165	184
$0.18728\,\beta\log(\beta) - 1.0192\,\beta + 16.10 + 7$	456	587	242	313	370
$0.000784314\,\beta^2 + 0.366078\,\beta - 6.125 + \log(8d) + 7$	535	722	270	350	410

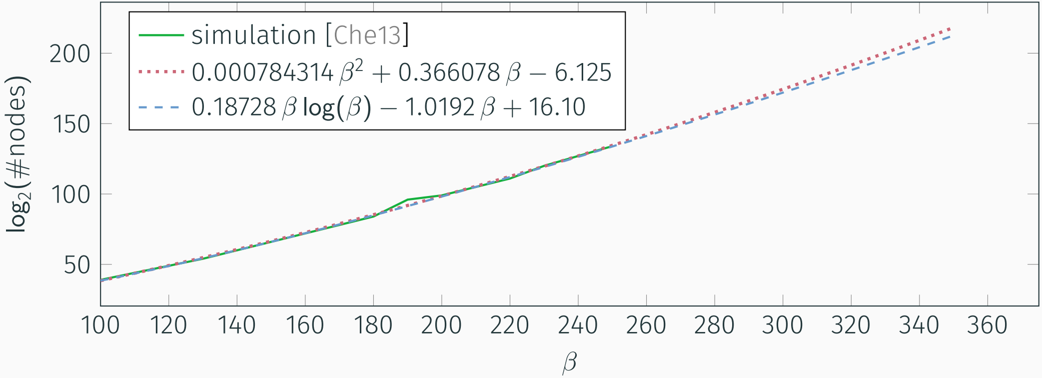
Sieving

- Produce new, shorter vectors by considering sums and differences of existing vectors
- **Time:** $2^{\mathcal{O}(\beta)}$
- **Memory:** $2^{\mathcal{O}(\beta)}$

Enumeration

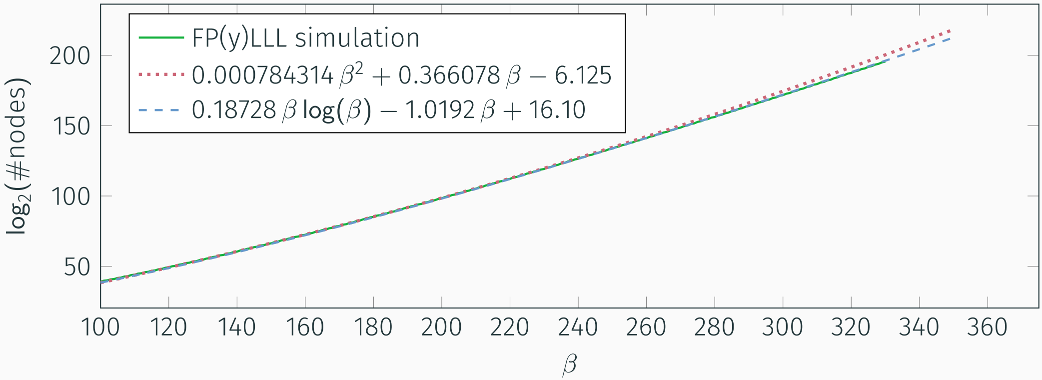
- Search through vectors smaller than a given bound: project down to 1-dim problem, lift to 2-dim problem …
- **Time:** $2^{\mathcal{O}(\beta\log\beta)}$ or $2^{\mathcal{O}(\beta^2)}$
- **Memory:** $\mathrm{poly}(\beta)$

Both estimates extrapolate the same data set

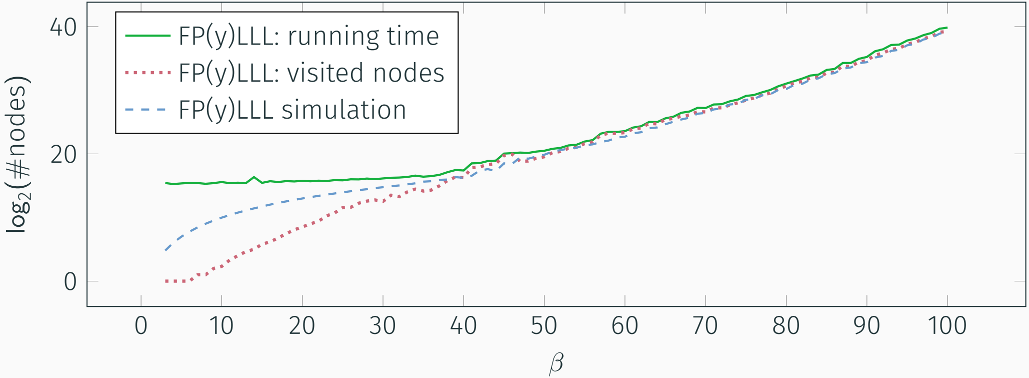


Extended Enumeration Simulation

Both estimates compared to our simulation



Enumeration Simulation vs Experiments



assuming 1 node ≈ 100 cpu cycles

Enumeration Wors-Case Complexity

Scheme / Cost Model	Kyber	Lima	R EMBLEM	NTRU HRSS	SNTRU'
0.292β	180	218	112	136	155
$0.292\beta + 16.4$	196	234	129	152	171
$0.292\beta + \log(8d) + 16.4$	210	248	142	165	184
$0.18728\,\beta\log(\beta) - 1.0192\,\beta + 16.10 + 7$	456	587	242	313	370
$0.000784314\,\beta^2 + 0.366078\,\beta - 6.125 + 7$	535	722	270	350	410

Known worst-case hardness of Kannan's enumeration is[7]

$$\beta^{1/(2e)\beta+o(\beta)} \approx \beta^{0.1839\,\beta+o(\beta)}$$

Open Question

Can we do better than worst-case hardness inside BKZ?

[7] Guillaume Hanrot and Damien Stehlé. Improved Analysis of Kannan's Shortest Lattice Vector Algorithm. In: *CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. LNCS. Springer, Heidelberg, Aug. 2007, pp. 170–186. DOI: 10.1007/978-3-540-74143-5_10.

Sieving vs Enumeration

Scheme / Cost Model	Kyber	Lima	R EMBLEM	NTRU HRSS	SNTRU'
0.292β	180	218	112	136	155
$0.292\beta + 16.4$	196	234	129	152	171
$0.292\beta + \log(8d) + 16.4$	210	248	142	165	184
$0.18728\,\beta\log(\beta) - 1.0192\,\beta + 16.10 + 7$	456	587	242	313	370
$0.000784314\,\beta^2 + 0.366078\,\beta - 6.125 + 7$	535	722	270	350	410

Sieving is asymptotically faster than enumeration, but does it beat enumeration in practical or cryptographic dimensions?

Sieving: G6K

G6K[8] is a Python/C++ framework for experimenting with sieving algorithms (inside and outside BKZ)

- Does not take the "oracle" view appealed to earlier but considers sieves as stateful machines.
- Implements several sieve algorithms[9] (but not the asymptotically fastest[10] ones)
- Applies many recent tricks and adds new tricks for improving performance of sieving

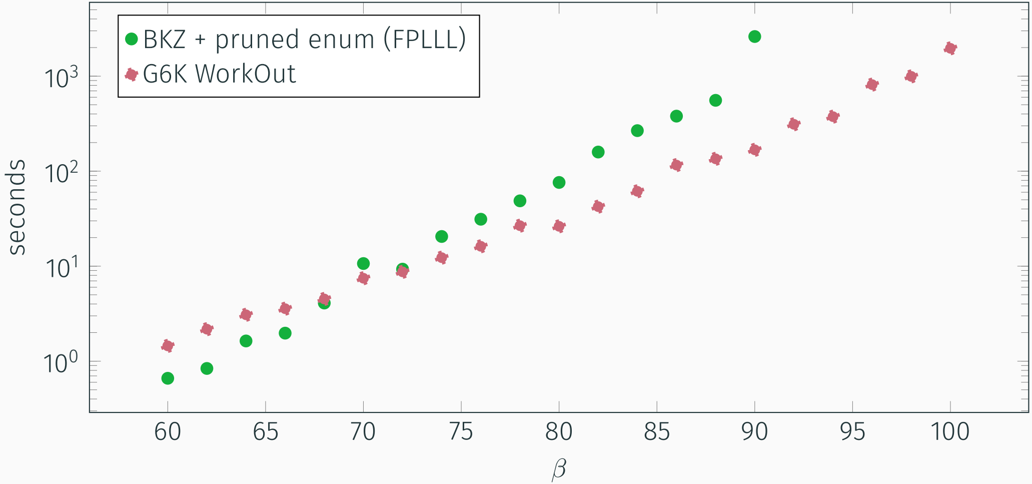
---

[8] 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. to appear. 2019.

[9] Gauss, NV, BGJ1 (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. `http://eprint.iacr.org/2015/522`. 2015; with one level of filtration)

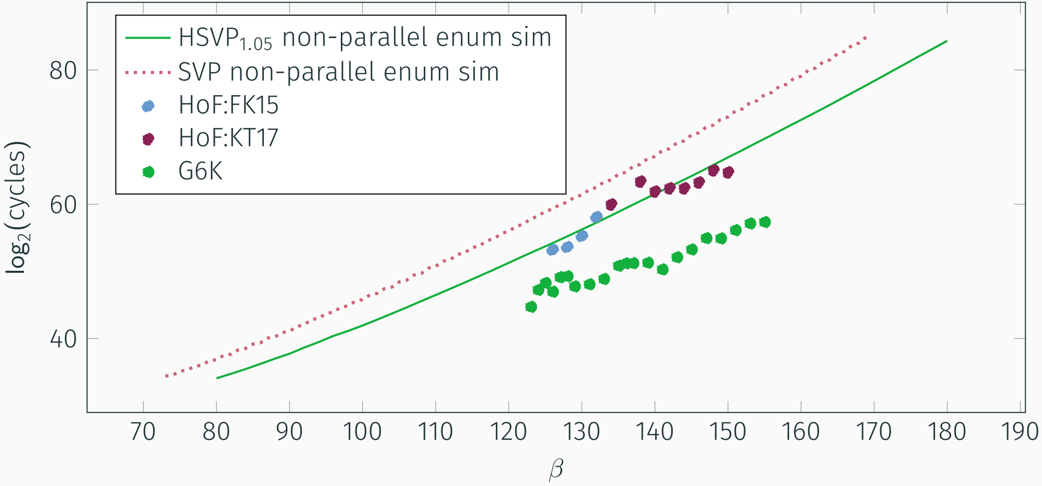
[10] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In: *27th SODA*. ed. by Robert Krauthgamer. ACM-SIAM, Jan. 2016, pp. 10–24. DOI: `10.1137/1.9781611974331.ch2`.

Sieving: SVP



Average time in seconds for solving exact SVP

Darmstadt HSVP$_{1.05}$ Challenges



Estimated and reported costs for solving Darmstadt SVP Challenges.

Sieving: Open Questions

- G6K does not support coarse grained parallelism across different machines yet: not clear how exponential memory requirement scales in this regime
- Practical performance of asymptotically faster sieves still unclear
- Dedicated hardware ...

Quantum Estimates

Type	Scheme / Cost Model	Kyber	Lima	R EMBLEM	NTRU HRSS	SNTRU'
classical	$0.292\beta + \log(8d) + 16.4$	210	248	142	165	184
quantum	$0.265\beta + \log(8d) + 16.4$	193	228	131	153	170
classical	$0.18728\,\beta\log(\beta) - 1.0192\,\beta + 16.10$	456	587	242	313	370
quantum	$1/2\,(0.18728\,\beta\log(\beta) - 1.0192\,\beta + 16.10)$	228	294	121	157	187

Sieving Given some vector $\mathbf{w}$ and a list of vectors L, apply Grover's algorithm to find $\{\mathbf{v} \in L \text{ s.t. } \|\mathbf{v} \pm \mathbf{w}\| \leq \|\mathbf{w}\|\}$.[11]

Enumeration Apply Montanaro's quantum backtracking algorithm for quadratic speed-up.[12]

[11] Thijs Laarhoven. Search problems in cryptography: From fingerprinting to lattice sieving. PhD thesis. Eindhoven University of Technology, 2015.

[12] Yoshinori Aono, Phong Q. Nguyen, and Yixin Shen. Quantum Lattice Enumeration and Tweaking Discrete Pruning. Cryptology ePrint Archive, Report 2018/546. `https://eprint.iacr.org/2018/546`. 2018.

Quantum Sieving

- A quantum sieve needs list of $2^{0.2075\beta}$ vectors before pairwise search with Grover
- Newer sieves use that the search is structured, Grover does unstructured search
  - Quantum Gauss Sieve

$$2^{(0.2075+\frac{1}{2}0.2075)\,\beta+o(\beta)} = 2^{0.311\,\beta+o(\beta)} \text{ time}, \qquad 2^{0.2075\,\beta+o(\beta)} \text{ memory}$$

  - Classical BGJ Sieve[13]

$$2^{0.311\,\beta+o(\beta)} \text{ time}, \qquad 2^{0.2075\,\beta+o(\beta)} \text{ memory}$$

- Asymptotically fastest sieves have small lists and thus less Grover speed-up potential

[13] 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. http://eprint.iacr.org/2015/522. 2015.

A Word on Lower Bounds

Type	Scheme / Cost Model	Kyber	Lima	R EMBLEM	NTRU HRSS	SNTRU'
classical	0.292β [Alk+16]	180	218	112	136	155
quantum	0.265β [Alk+16]	163	198	102	123	140
classical	$0.123\,\beta\log(\beta) - 0.70\beta + 6.1$ [Aon+18]	276	358	142	186	224
quantum	$0.061\,\beta\log(\beta) - 0.35\beta + 2.6$ [Aon+18]	135	175	69	91	109

These estimates ignore:

- (large) polynomial factors hidden in $o(\beta)$
- MAXDEPTH of quantum computers
- cost of a Grover iteration

Thus:

- cannot claim parameters need to be adjusted when these estimates are lowered
- must be careful about conclusions drawn in these models: some attacks don't work here but work in reality

More Open Questions

- Many submissions use small and sparse secrets where combinatorial techniques apply. Cost of these not fully understood.
- (Structured) Ideal-SVP is easier than General SVP on a quantum computer.[14] Ring-LWE (but for a choice of parameters typically not used in practice) is at least as hard as Ideal-SVP, but it is not clear if it is harder, e.g. if those attacks extend.
- The effect of decryption failures in probabilistic encryption based on LWE not fully understood. Some submissions completely eliminate these.

[14] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP. In: *EUROCRYPT 2017, Part I*. ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Heidelberg, 2017, pp. 324–348. DOI: `10.1007/978-3-319-56620-7_12`.

Fin

Thank You

[Alb+17] 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.

[Alb+18] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate All the LWE, NTRU Schemes! In: *SCN 18*. Ed. by Dario Catalano and Roberto De Prisco. Vol. 11035. LNCS. Springer, Heidelberg, Sept. 2018, pp. 351–367. DOI: 10.1007/978-3-319-98113-0_19.

[Alb+19] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The General Sieve Kernel and New Records in Lattice Reduction. to appear. 2019.

[Alk+16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum Key Exchange - A New Hope. In: *25th USENIX Security Symposium, USENIX Security 16*. Ed. by Thorsten Holz and Stefan Savage. USENIX Association, 2016, pp. 327–343. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim.

[ANS18] Yoshinori Aono, Phong Q. Nguyen, and Yixin Shen. Quantum Lattice Enumeration and Tweaking Discrete Pruning. Cryptology ePrint Archive, Report 2018/546. https://eprint.iacr.org/2018/546. 2018.

[Aon+18] Yoshinori Aono, Phong Q. Nguyen, Takenobu Seito, and Junji Shikata. Lower Bounds on Lattice Enumeration with Extreme Pruning. In: *CRYPTO 2018, Part II*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10992. LNCS. Springer, Heidelberg, Aug. 2018, pp. 608–637. DOI: 10.1007/978-3-319-96881-0_21.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

[Bec+16] 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. http://eprint.iacr.org/2015/522. 2015.

[CDW17] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP. In: *EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Heidelberg, 2017, pp. 324–348. DOI: 10.1007/978-3-319-56620-7_12.

[Che13] Yuanmi Chen. Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe. PhD thesis. Paris 7, 2013.

[Hof+15] Jeff Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, William Whyte, and Zhenfei Zhang. Choosing Parameters for NTRUEncrypt. Cryptology ePrint Archive, Report 2015/708. http://eprint.iacr.org/2015/708. 2015.

[HS07] Guillaume Hanrot and Damien Stehlé. Improved Analysis of Kannan's Shortest Lattice Vector Algorithm. In: *CRYPTO 2007*. Ed. by Alfred Menezes. Vol. 4622. LNCS. Springer, Heidelberg, Aug. 2007, pp. 170–186. DOI: `10.1007/978-3-540-74143-5_10`.

[Laa15] Thijs Laarhoven. Search problems in cryptography: From fingerprinting to lattice sieving. PhD thesis. Eindhoven University of Technology, 2015.

[LN13] Mingjie Liu and Phong Q. Nguyen. Solving BDD by Enumeration: An Update. In: *CT-RSA 2013*. Ed. by Ed Dawson. Vol. 7779. LNCS. Springer, Heidelberg, 2013, pp. 293–309. DOI: `10.1007/978-3-642-36095-4_19`.

[Sma+17] Nigel P. Smart, Martin R. Albrecht, Yehuda Lindell, Emmanuela Orsini, Valery Osheter, Kenny Paterson, and Guy Peer. LIMA. Tech. rep. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`. National Institute of Standards and Technology, 2017.