

Fast Practical Lattice Reduction through Iterated Compression

Keegan Ryan and Nadia Heninger

Solving the RSA partial factorization problem

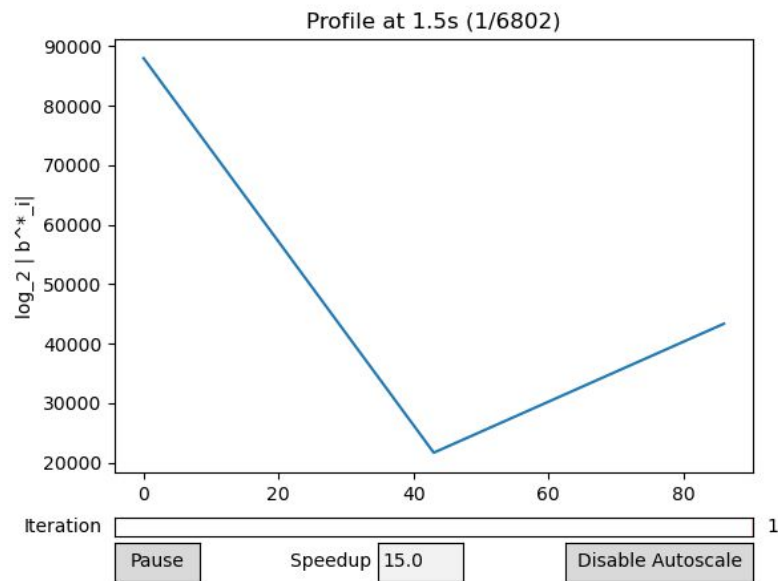
Given a 2048-bit RSA modulus $N = pq$ and 512 most significant bits of p , factor N .

Previous results*:

470 core-hours

Our results:

~18 core-hours



*from [AHMP23], <https://eprint.iacr.org/2023/329>

Solving the Gentry-Halevi FHE problem

Given a public key for the Gentry-Halevi FHE scheme, recover the private key.

Previous results (toy/small/medium)*:

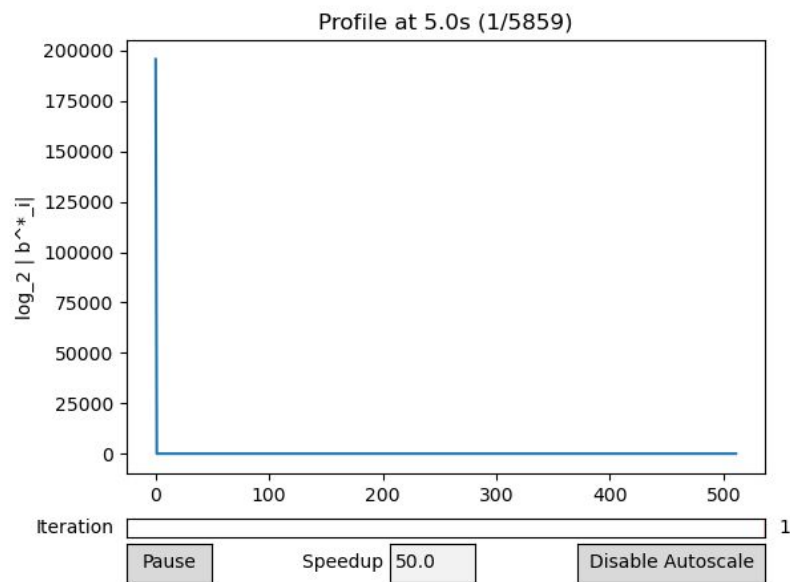
24 core-days/15.7 core-years/

68582 core-years

Our results:

15 core-minutes/31 core-hours/

6.4 core-years

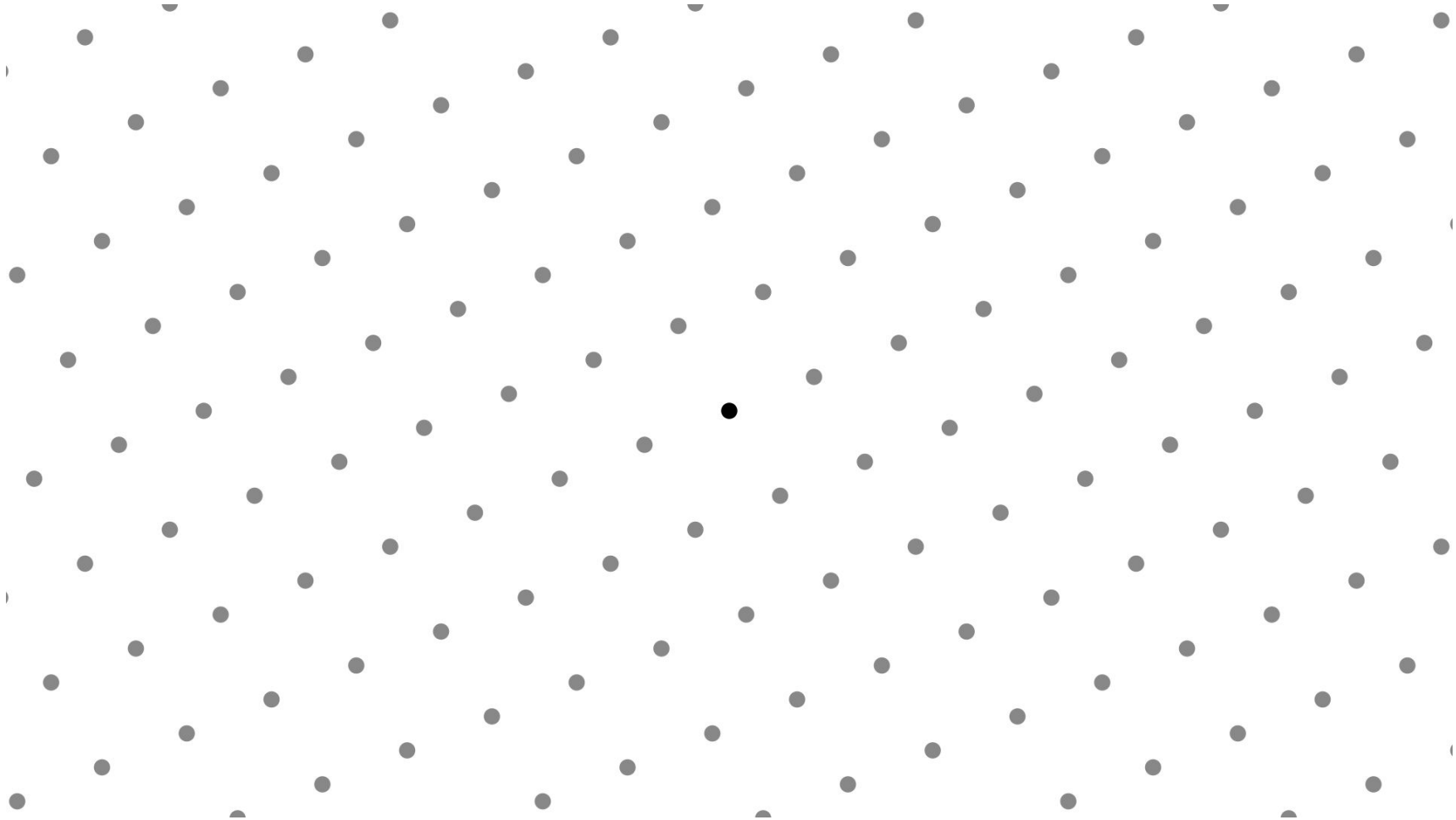


*from [PSZ15], <https://doi.org/10.1007/s10623-014-9957-1>

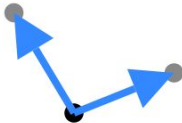
Talk Outline

- What is lattice reduction?
 - Geometric intuition for lattices
 - Profile-based intuition for reduction
- How do we improve lattice reduction from [LLL82]?
 - Recursion [KS01, NS16, KEF21]
 - Floating-point numbers [NS09, KEF21]
- How do we merge both strategies?
 - Basis compression [SMSV14] and profile drop
 - Algorithm runtime analysis
- Results

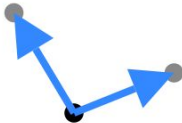
What is Lattice Reduction?



$$\begin{bmatrix} 5 \\ 2 \end{bmatrix} \begin{bmatrix} -3 \\ 5 \end{bmatrix}$$



$$\begin{bmatrix} 5 & -3 \\ 2 & 5 \end{bmatrix}$$



$$\begin{bmatrix} 31 & 18 \\ 0 & 1 \end{bmatrix}$$



Geometric and Computational Interpretation

- Lattices are mathematical objects represented by a basis matrix.
- Many cryptanalytic problems can be solved using lattices:
 - Find an integer linear combination of polynomials with small coefficients -> Determine a short vector in a lattice
 - Decrypt an FHE ciphertext -> Find distance to the closest lattice point
- Our ability to solve the lattice problem depends on the quality of the basis

Good vs. Bad Bases

$$\begin{bmatrix} 29 & 11 \\ 0 & 2 \end{bmatrix}$$

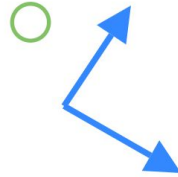


Does the lattice generated by this basis include a vector within the green circle?

Unclear.

Good vs. Bad Bases

$$\begin{bmatrix} 4 & 7 \\ 6 & -4 \end{bmatrix}$$

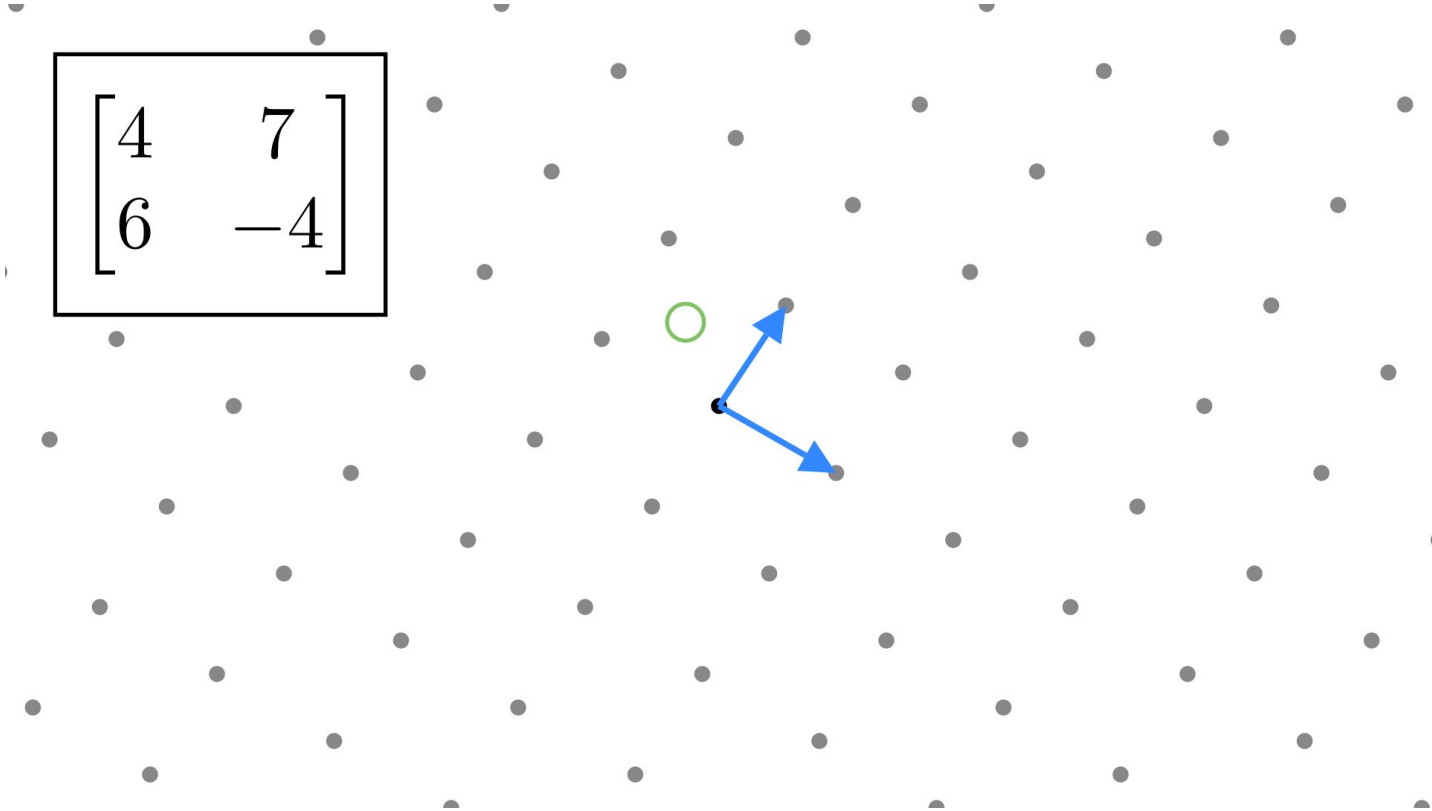


Does the lattice generated by this basis include a vector within the green circle?

No.

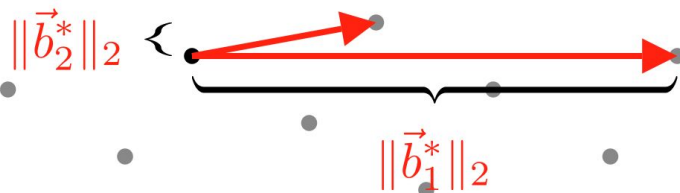
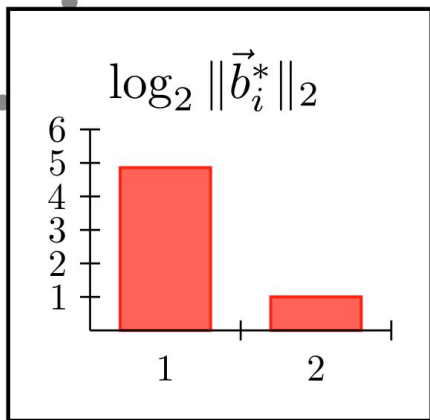
Good vs. Bad Bases

$$\begin{bmatrix} 4 & 7 \\ 6 & -4 \end{bmatrix}$$



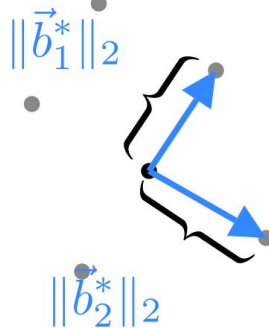
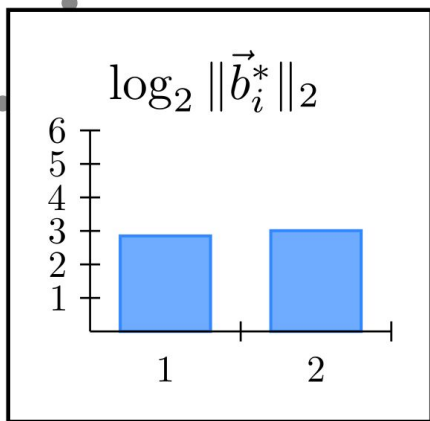
Good vs. Bad Bases

$$\begin{bmatrix} 29 & 11 \\ 0 & 2 \end{bmatrix}$$



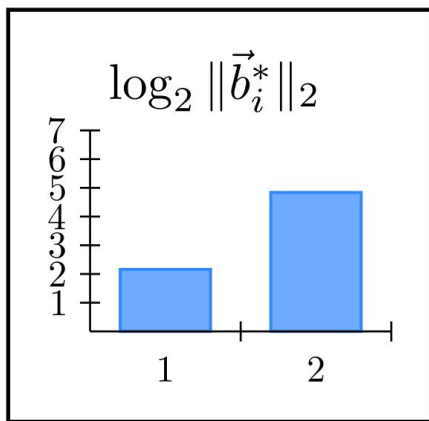
Good vs. Bad Bases

$$\begin{bmatrix} 4 & 7 \\ 6 & -4 \end{bmatrix}$$



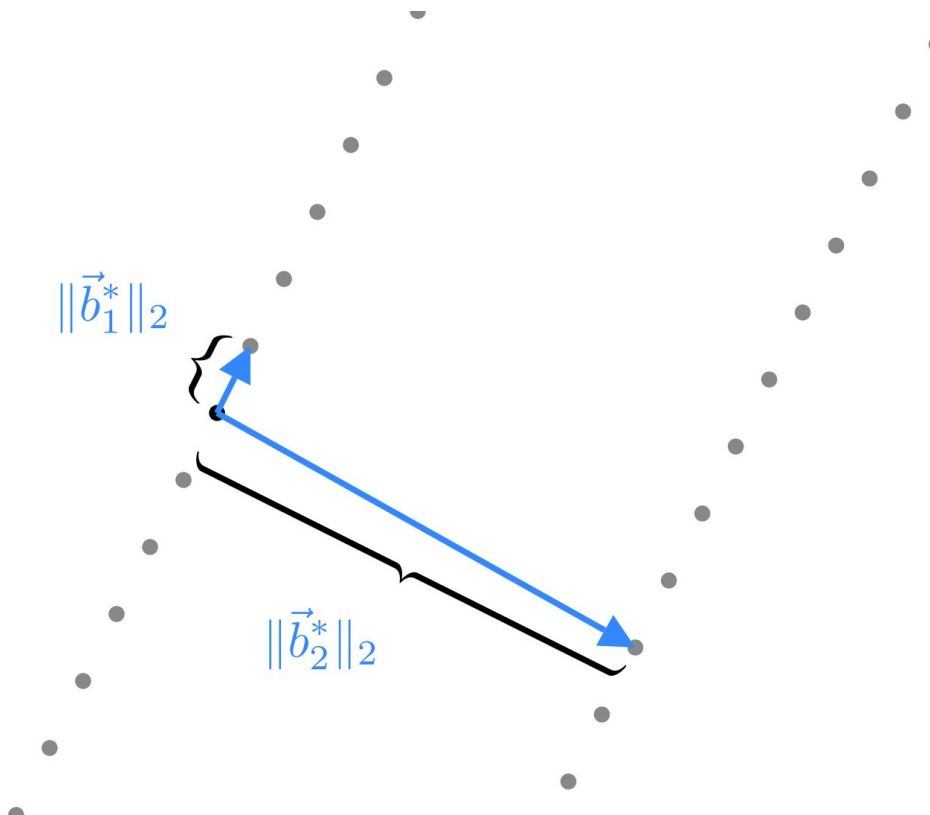
Good vs. Bad Bases

$$\begin{bmatrix} 2 & 25 \\ 4 & -14 \end{bmatrix}$$



$$\|\vec{b}_1^*\|_2$$

$$\|\vec{b}_2^*\|_2$$



Lattice Reduction

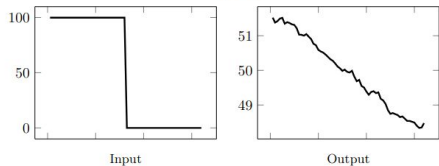
- A lattice reduction algorithm takes a lattice basis as input and returns a “good” basis of the same lattice.
- “Good” can mean many things, but one definition is the Lovász condition:

$$\log \|b_i^*\| - \log \|b_{i+1}^*\| \leq -\frac{1}{2} \log(\delta - \mu_{i+1,i}^2) < 0.21$$

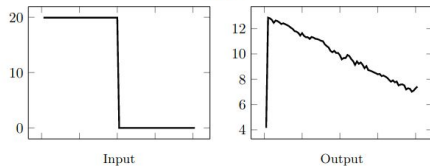
- Intuitively,
 - Large decreases in the profile are disallowed
 - Small decreases or large increases in the profile are OK

Input and Output bases

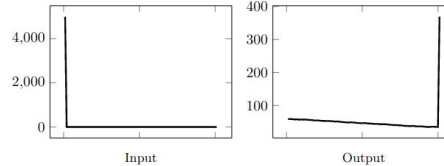
Q-ary



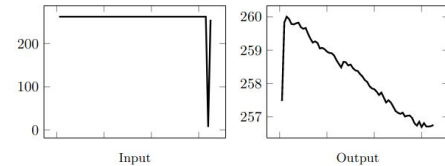
LWE



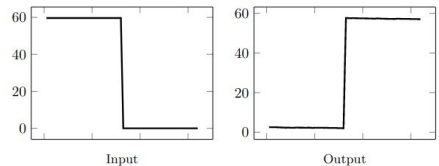
Approximate GCD



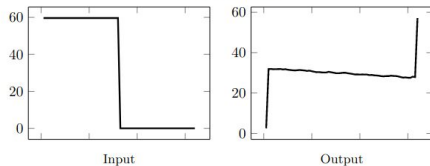
HNP



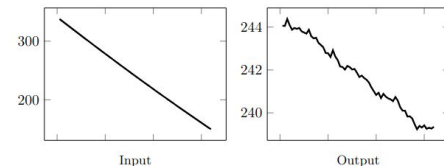
NTRU



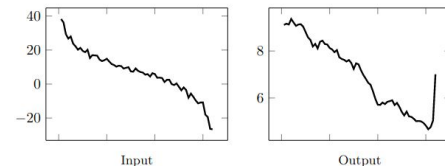
NTRU-like



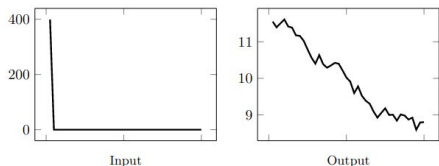
Ajtai



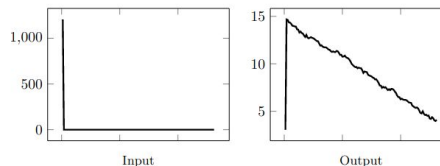
GGH



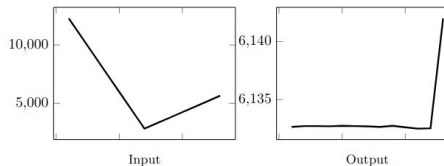
Goldstein-Mayer



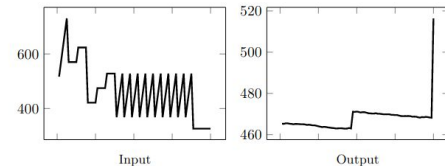
Knapsack



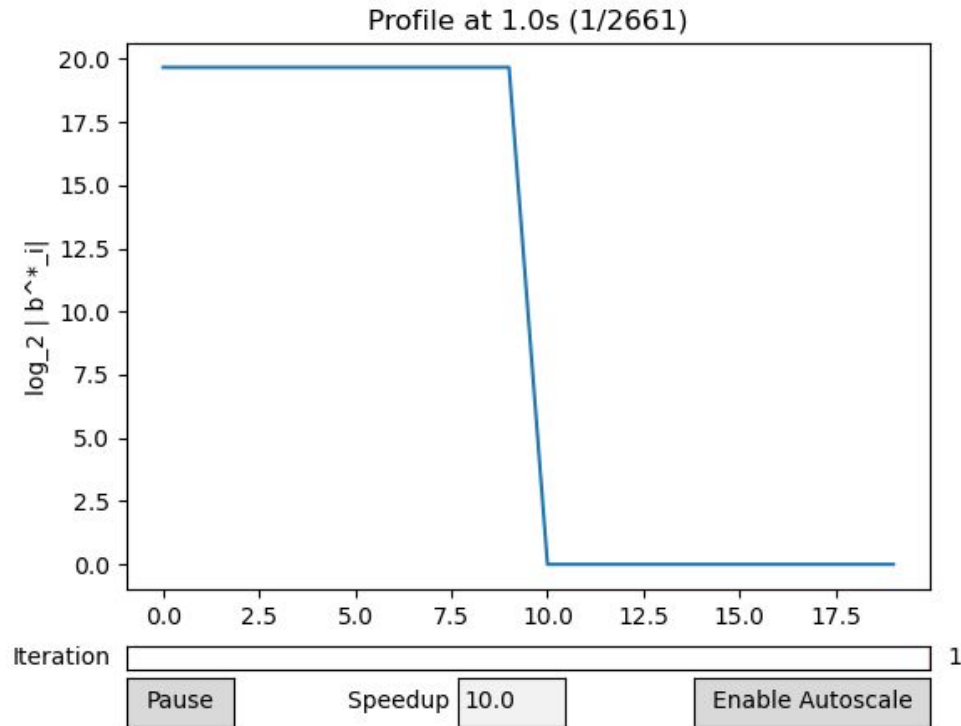
RSA Partial Factorization



ECHNP



[LLL82] lattice reduction algorithm



Improving Lattice Reduction

Problems with [LLL82]

- Polynomial time in dimension and bitsize of entries:

$$O(n^{5+\varepsilon}(p + \log n)^{2+\varepsilon})$$

- Hard to reduce larger than dimension 20
- Computing the Gram-Schmidt orthogonalization is *expensive*
 - Number of times the GSO is updated: $O(n^2(p + \log n))$
 - Number of arithmetic operations per update: $O(n^2)$
 - Bit sizes of rational numbers in the GSO: $O(n(p + \log n))$

Problems with [LLL82]

- Polynomial time in dimension and bitsize of entries:

$$O(n^{5+\varepsilon}(p + \log n)^{2+\varepsilon})$$

- Hard to reduce larger than dimension 20
- Computing the Gram-Schmidt orthogonalization is *expensive*

- Number of times the GSO is updated:
- Number of arithmetic operations per update:
- Bit sizes of rational numbers in the GSO:

$$O(n^2(p + \log n))$$

← Recursion

$$O(n^2)$$

$$O(n(p + \log n))$$

← Precision

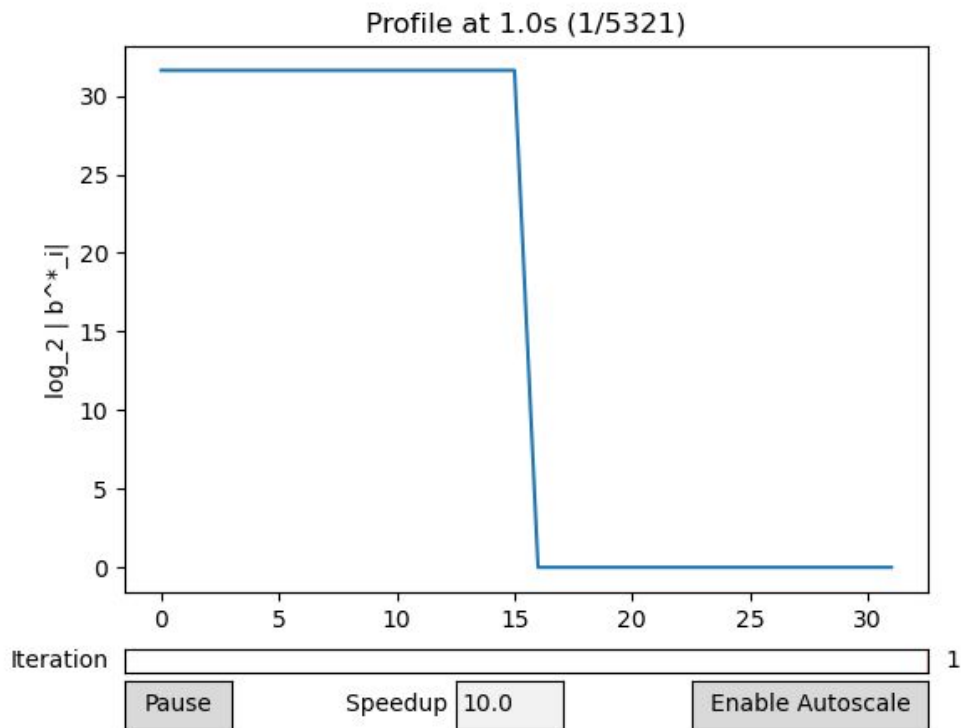
Recursive Approach: [KS01, NS16]

- Use upper-triangular GSO to describe *projected sublattices*

$$\begin{bmatrix} 15595 & 2453 & 1718 & -6725 \\ 0 & 267 & -128 & 106 \\ 0 & 0 & 23 & -4 \\ 0 & 0 & 0 & 7 \end{bmatrix} \longrightarrow \begin{bmatrix} 267 & -128 \\ 0 & 23 \end{bmatrix}$$

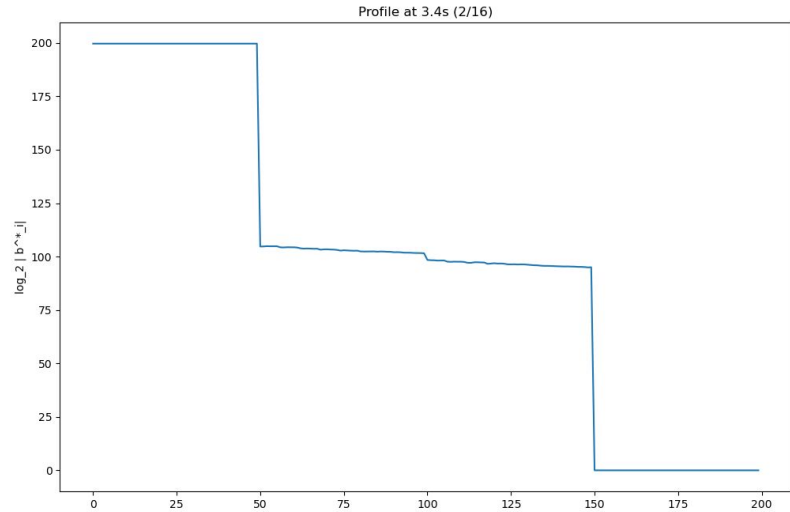
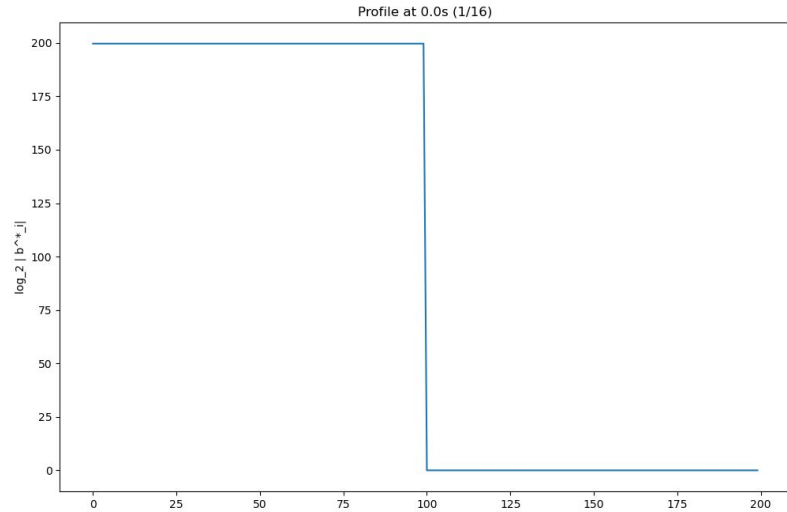
- Recursively reduce projected sublattices
- “Batches” updates to GSO, so fewer updates at large dimension
- [NS16] has best proven running time $O(n^{4+\varepsilon}(p + \log n)^{2+\varepsilon})$
 - Number of arithmetic operations to update GSO $O(n^{3+o(1)}(p + \log n)^{o(1)})$
 - Bit sizes of entries in the GSO $O(n(p + \log n))$
- Output may not satisfy the Lovász condition

[NS16] lattice reduction algorithm



Recursive Approach

We can apply the recursive approach when we have fully computed the GSO



Precision Management [NS09,KEF21]

- Instead of updating the GSO during recursive reduction, we can compute the GSO with lower precision
- How much precision is needed?

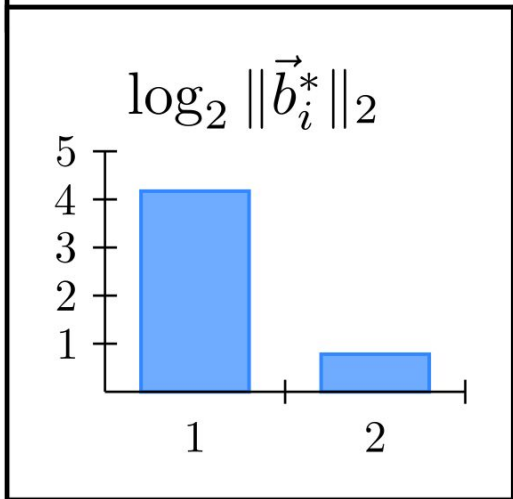
$$\begin{bmatrix} 18 & 5 \\ 1 & 2 \end{bmatrix}$$



$$\begin{bmatrix} \sqrt{325} & \frac{92}{325} \sqrt{325} \\ 0 & \sqrt{\frac{961}{325}} \end{bmatrix}$$



$$\begin{bmatrix} \sqrt{325} & \frac{92}{325} \sqrt{325} \\ 0 & \sqrt{\frac{961}{325}} \end{bmatrix}$$



$$\|\vec{b}_2^*\|_2$$



$$\|\vec{b}_1^*\|_2$$

$$\begin{bmatrix} \sqrt{325} & \frac{92}{325} \sqrt{325} \\ 0 & \sqrt{\frac{961}{325}} \end{bmatrix}$$

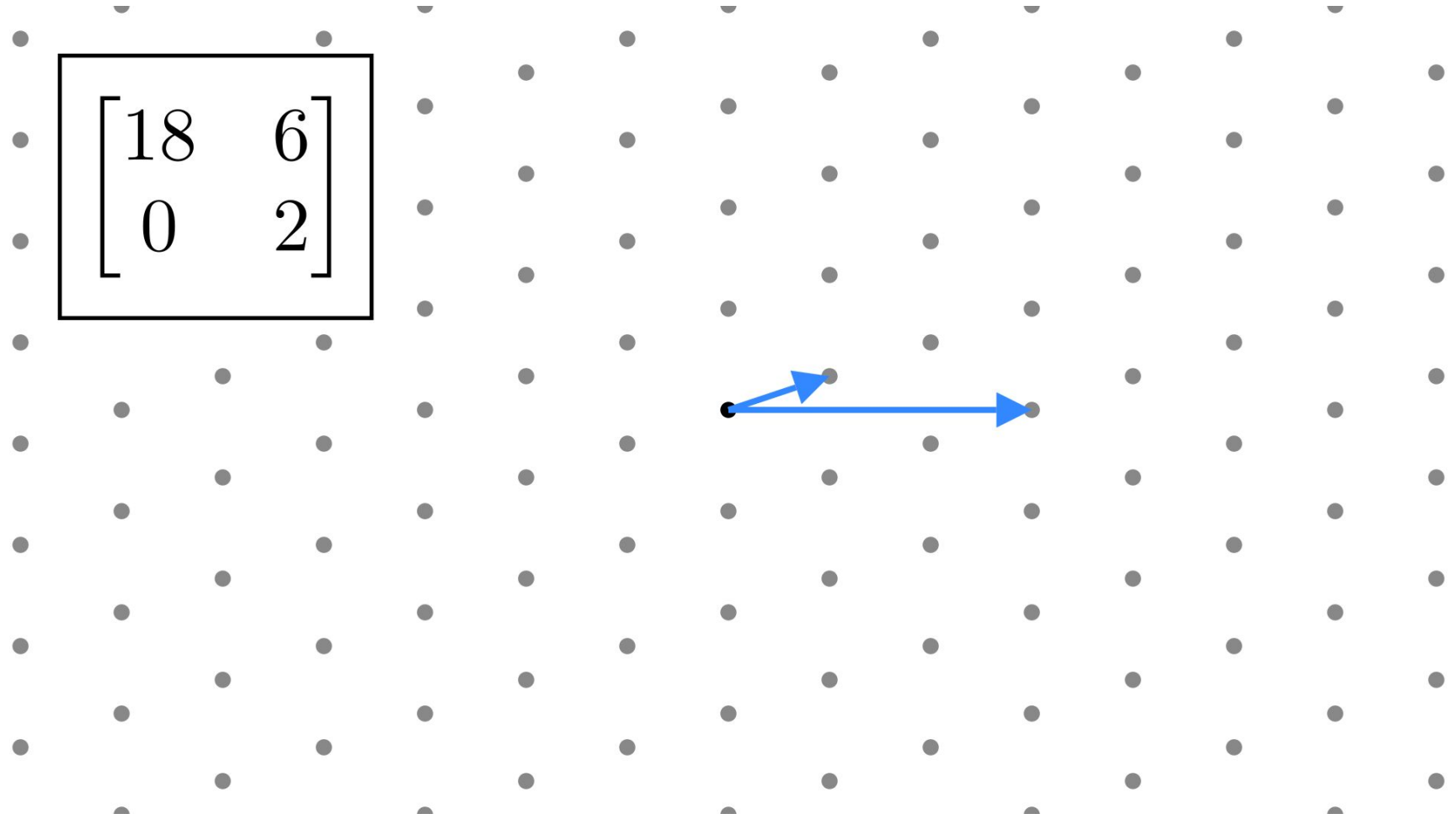


$$\begin{bmatrix} 18.0 & 5.1 \\ 0.0 & 1.7 \end{bmatrix}$$



$$\begin{bmatrix} 18 & 5 \\ 0 & 2 \end{bmatrix}$$





A 10x10 grid of gray dots is shown. A black rectangular box is positioned in the upper-left quadrant, containing a 2x2 matrix of numbers. The matrix is:

$$\begin{bmatrix} 18 & 6 \\ 0 & 2 \end{bmatrix}$$



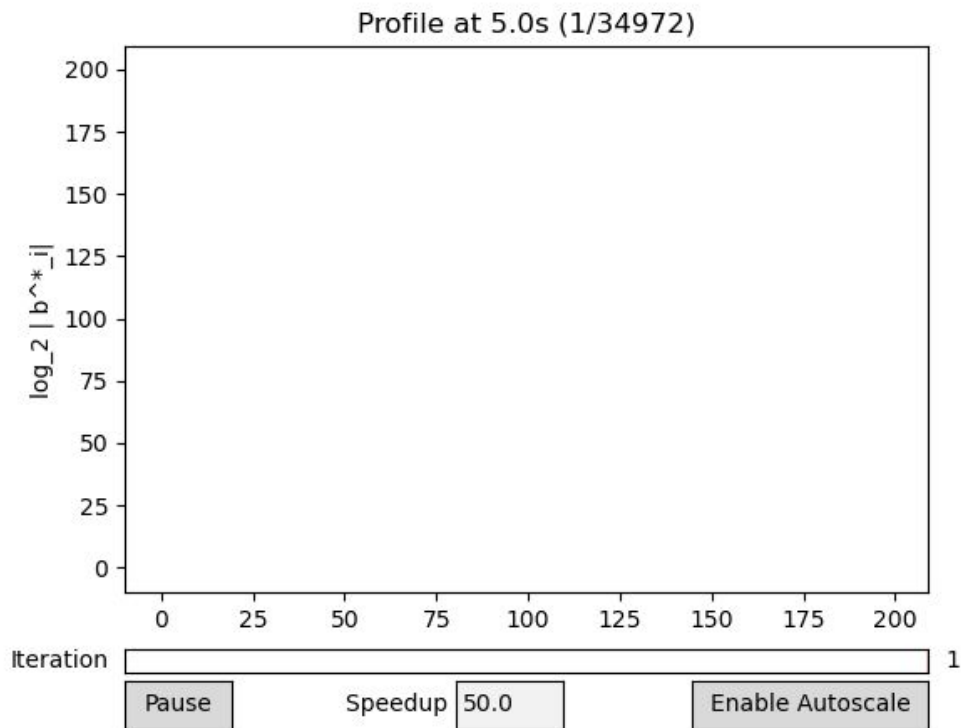
$$\begin{bmatrix} 20 & 4 \\ 0 & 0 \end{bmatrix}$$



Precision Management [NS09]

- We can round the exact representation without distorting the geometric properties too much.
- If we round too aggressively, the algorithm is unstable.
- [NS09] ensures stability by only computing the *prefix* of the GSO that already satisfies the Lovász condition
- Precision depends on the length of the prefix
 - Double precision fine up to dimension 170 [S09]
- [NS09] has proven running time
 - Number of rounds to update GSO $O(n^{4+\varepsilon}(p + \log n)(p + n))$
 - Bit cost to update GSO $O(n^2(p + \log n))$
 - Bit cost to update GSO $O(n^{2+\varepsilon}(p + n))$

[NS09] lattice reduction algorithm

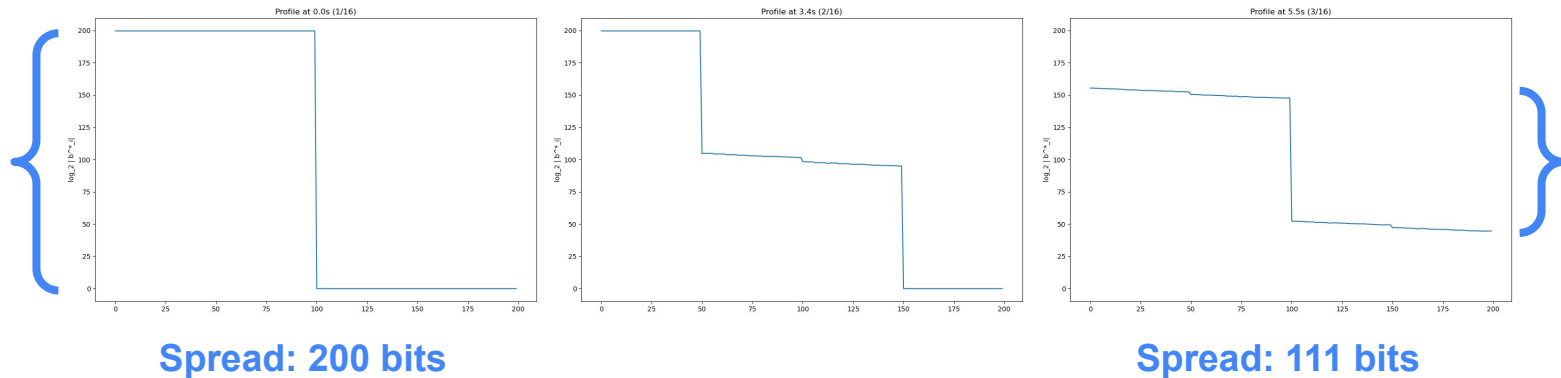


Precision Management [NS09]

- Computing the GSO is significantly faster in practice
- Used by FPLLL for bases of dimension ~ 300
- Incompatible with recursion, because GSO is only partially known

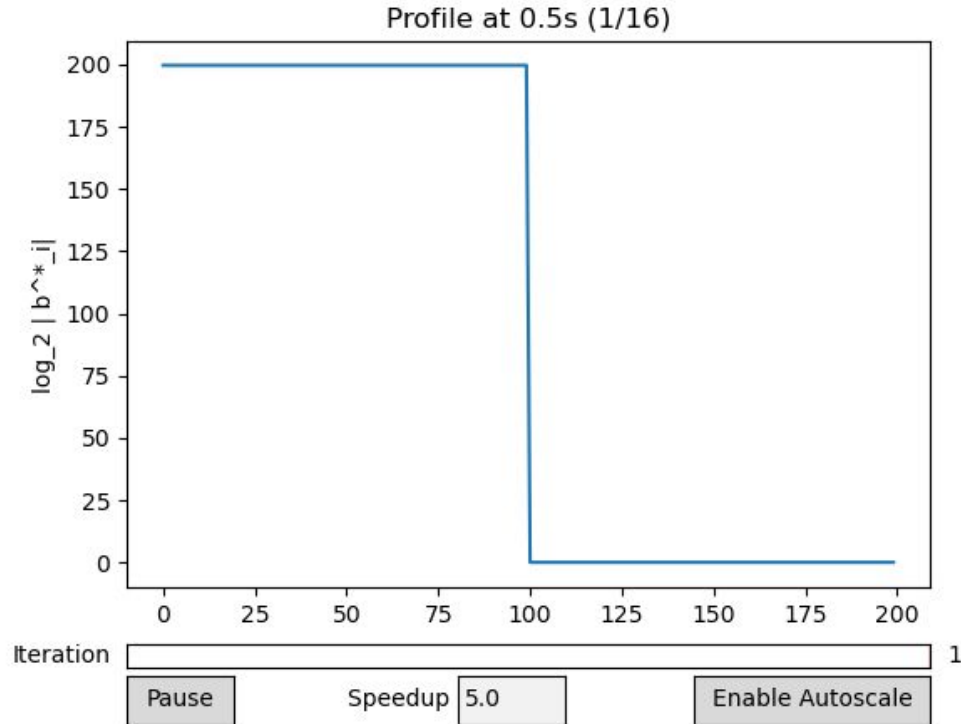
Precision+Recursion in [KEF21]

- Bits of precision depends on the vertical “spread” of the profile
- Recursive reduction heuristically decreases the spread exponentially quickly



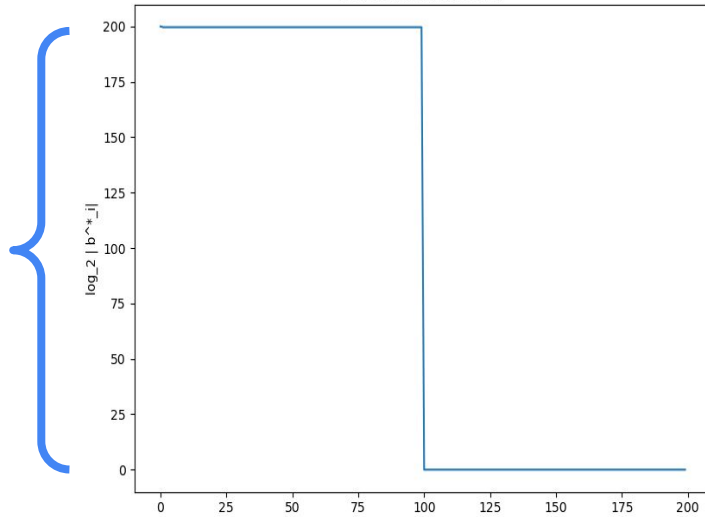
- Use fast matrix operations to update floating-point GSO
- Cost per update: $\tilde{O}(n^\omega p_i)$
- Total cost: $\tilde{O}\left(n^\omega p + n^\omega \frac{p}{2} + n^\omega \frac{p}{4} + \dots\right) = \tilde{O}(n^\omega p)$

[KEF21] lattice reduction algorithm

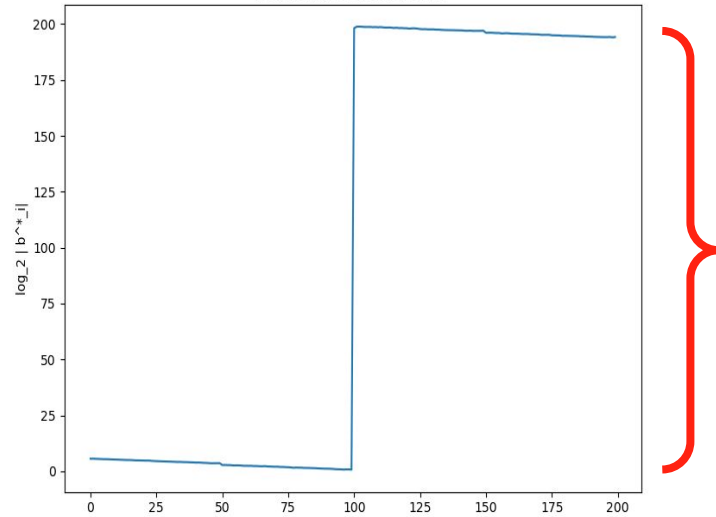


The heuristic assumption of [KEF21]
is wrong.

[KEF21] counterexample: NTRU bases



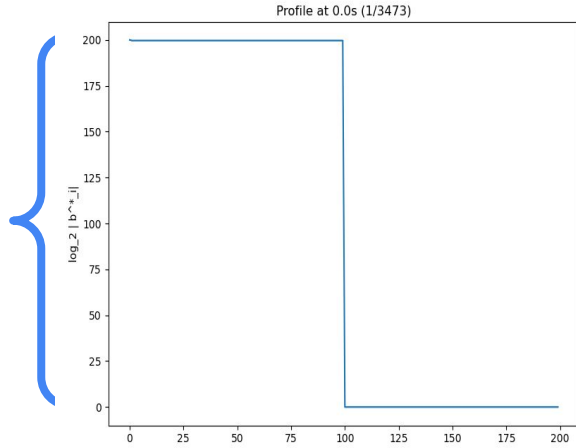
Spread: 200 bits



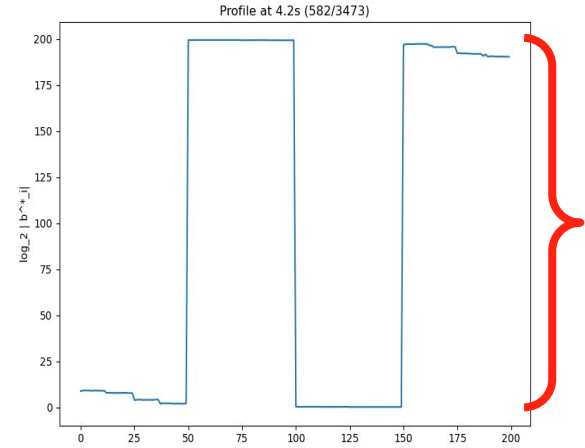
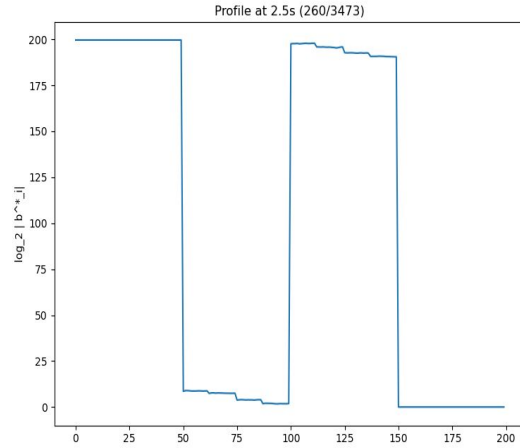
Spread: 198 bits

The spread might never decrease, so we can't rely on updates to the GSO to get less expensive over time.

[KEF21] counterexample: NTRU bases



Spread: 200 bits



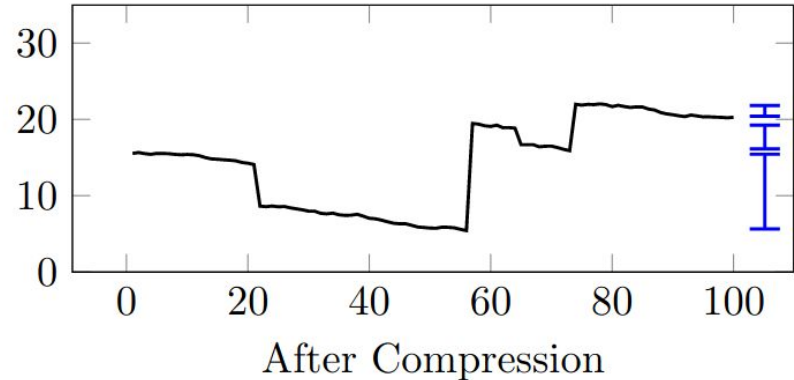
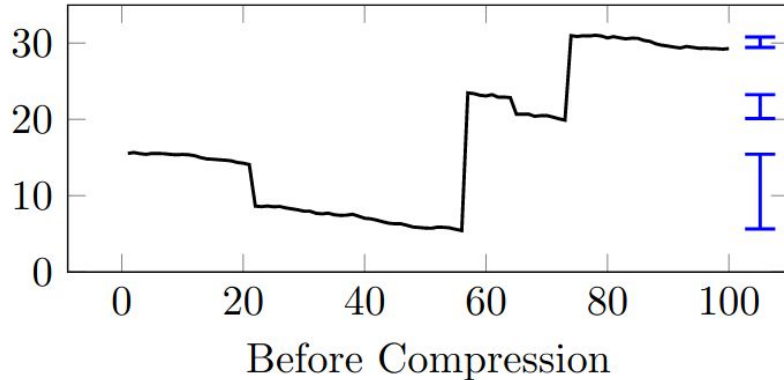
Spread: 200 bits

Special structure in the lattice can violate the heuristic assumption, so more bits are needed than predicted for stability.

[SMSV14] compression

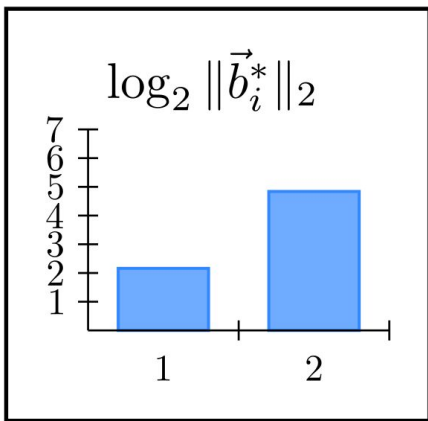
When we have a sparse projected sublattice orthogonal to a dense sublattice, we can scale down the projected sublattice without affecting lattice reduction.

Equivalently, when we have a large, sustained increase in the profile, we can shift the right side down to be closer to the left side.



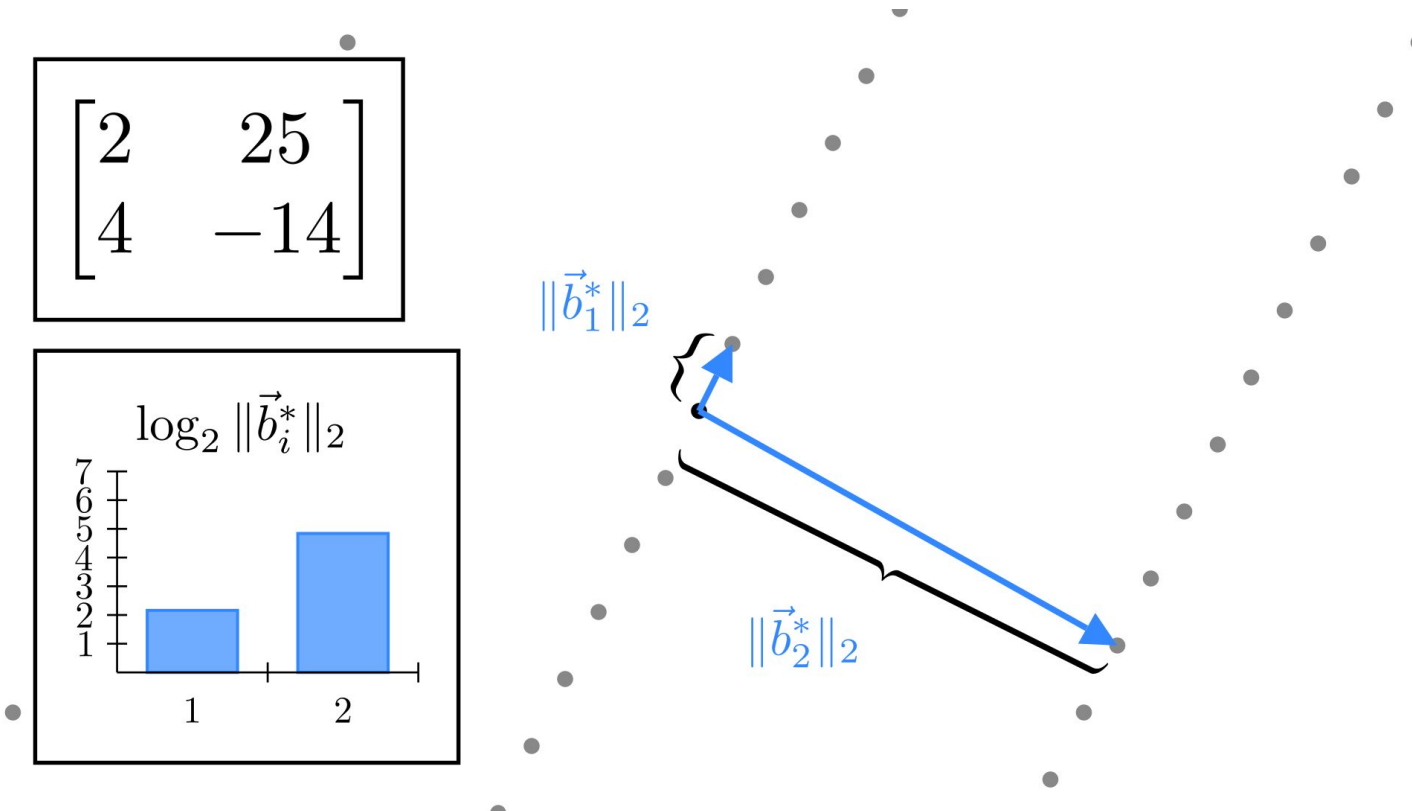
[SMSV14] compression

$$\begin{bmatrix} 2 & 25 \\ 4 & -14 \end{bmatrix}$$



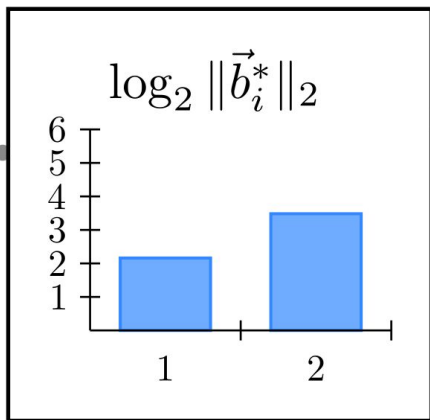
$$\|\vec{b}_1^*\|_2$$

$$\|\vec{b}_2^*\|_2$$



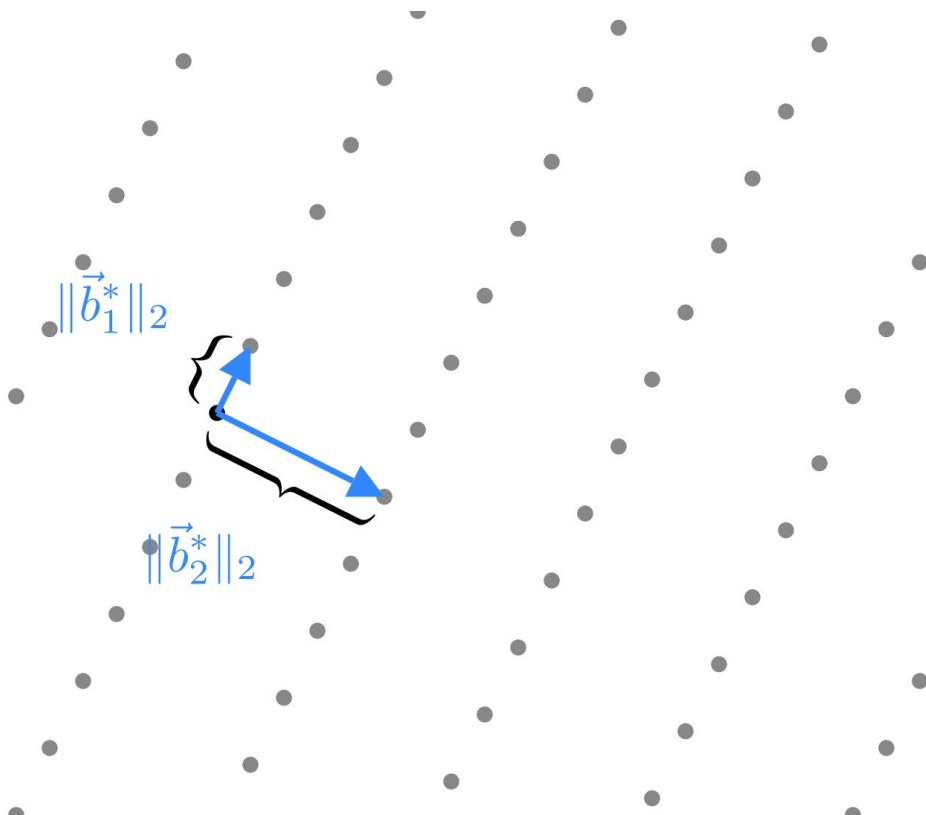
[SMSV14] compression

$$\begin{bmatrix} 2 & 10 \\ 4 & -5 \end{bmatrix}$$



$$\|\vec{b}_1^*\|_2$$

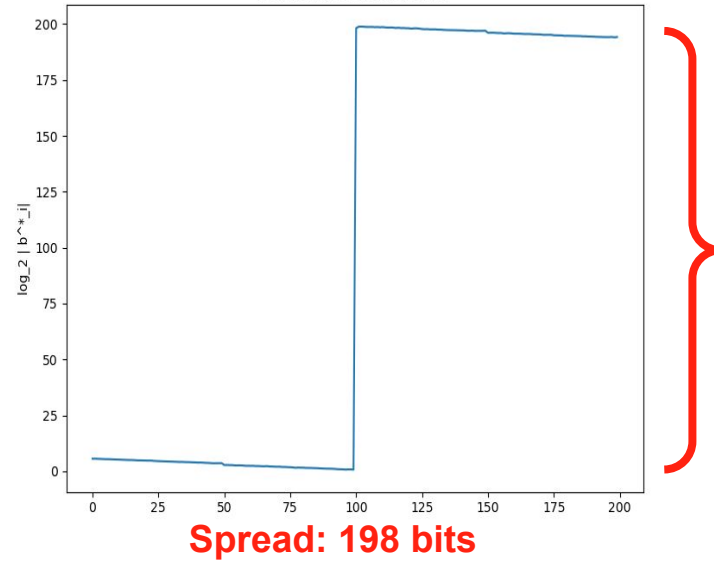
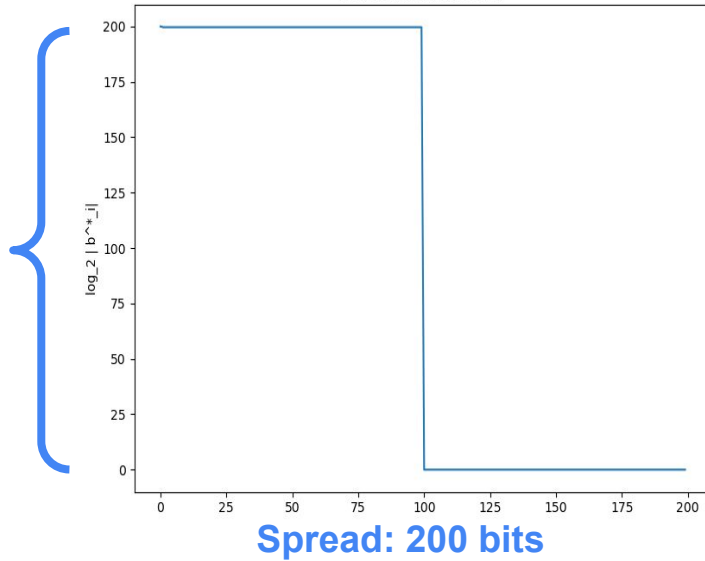
$$\|\vec{b}_2^*\|_2$$



Our improvements to [SMSV14] compression

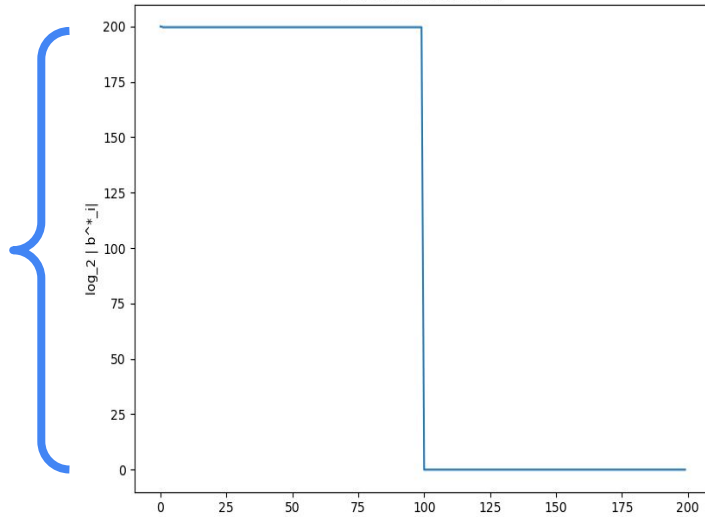
- We use the scaling technique to decrease the spread while maintaining necessary lattice geometry.
- Smaller spread means fewer bits of precision are necessary
- Unlike [SMSV14], we compress repeatedly, so we need to show stability.
- We define the *drop* as the spread of the compressed basis.
- Bounded drop is just as good as the Lovász condition
 - Small drop means no large decreases in the profile
 - Prove equivalent properties based on this condition
 - More compatible with recursive structure

Spread vs. drop of NTRU bases

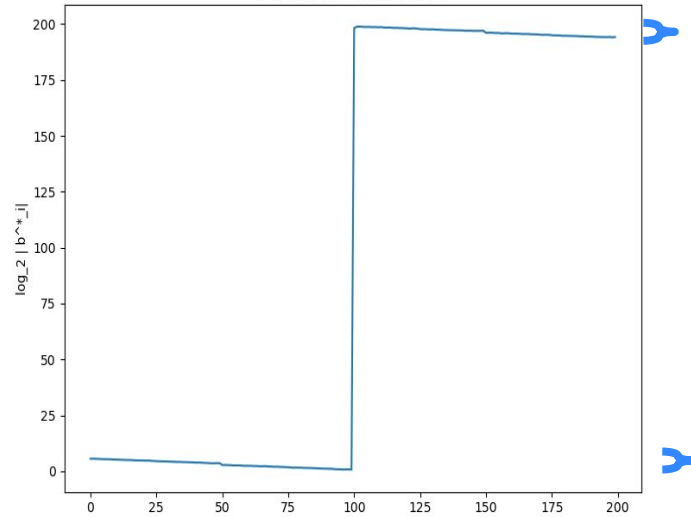


After lattice reduction, the spread of the lattice basis is large, but the drop is small.

Spread vs. drop of NTRU bases



Spread: 200 bits
Drop: 200 bits

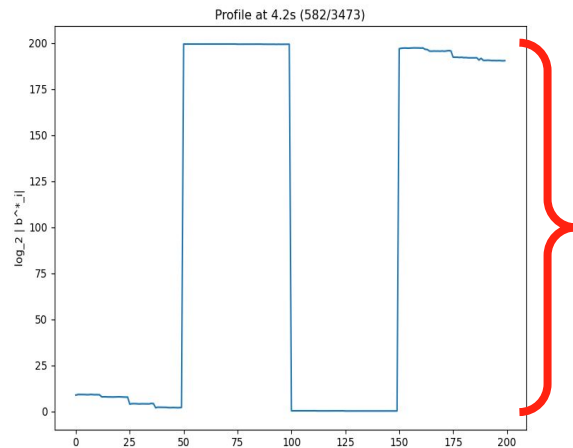
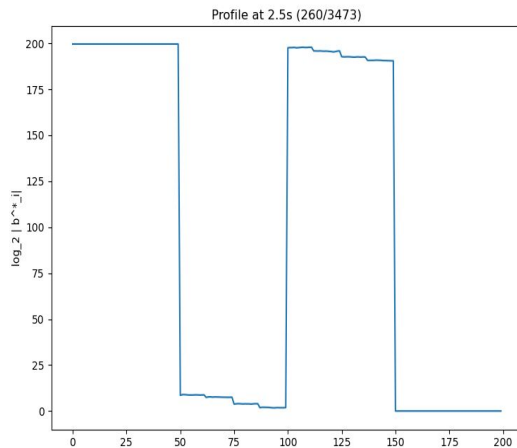
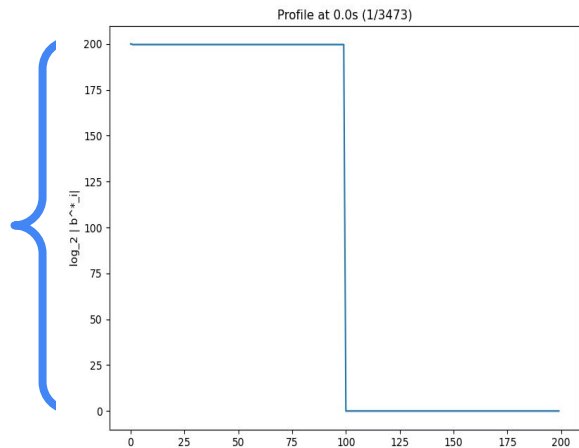


Spread: 198 bits
Drop: 10 bits

After lattice reduction, the spread of the lattice basis is large, but the drop is small.

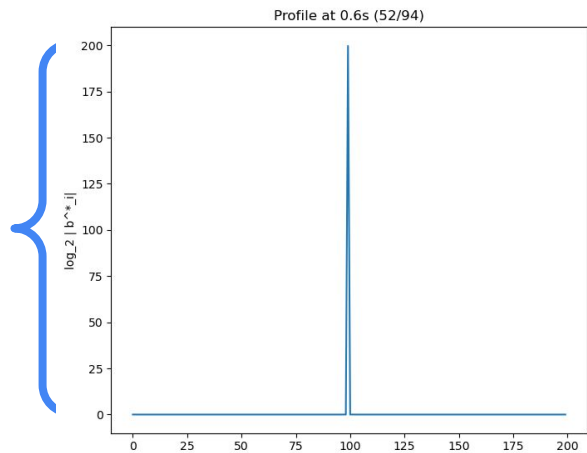
Running time

- Following [KEF21], can we prove that the *drop* decreases exponentially quickly with every round of recursive reduction? **No.**

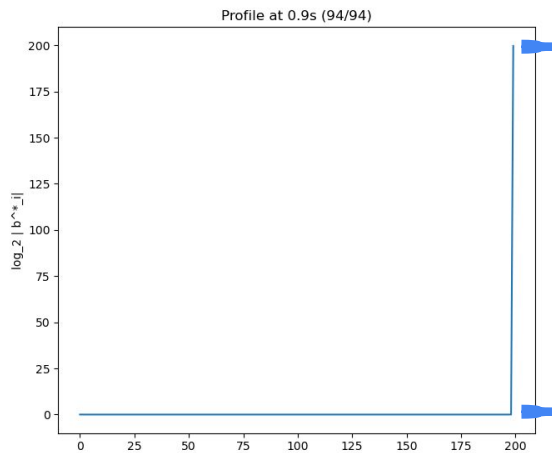
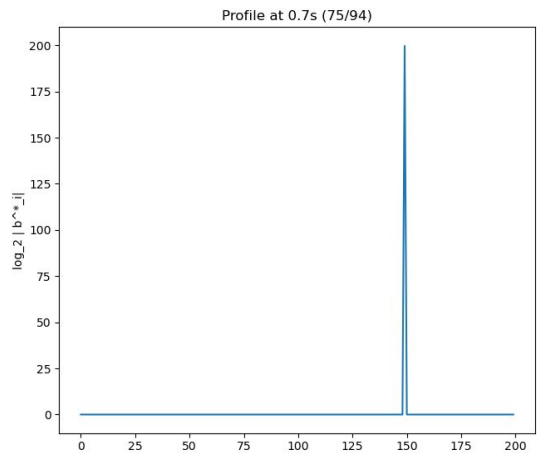


Running time

- Following [LLL82], can we prove that the decrease in *potential* is proportional to the working precision in every round? **No.**



Drop: 200 bits



Drop: 0 bits

Running time

- Can we prove that in every round, *one* of the two is true? **Yes***.
 - Either the decrease in drop is large, so future GSO updates require fewer bits of precision,
 - Or the decrease in potential is large, so we are significantly closer to being reduced.
- Cost per update: $O(n^\omega p_i^{1+\varepsilon})$
- Total cost**: $O(n^\omega (p + n)^{1+\varepsilon})$

*depends on heuristic assumptions about the final rounds

**for well-conditioned bases, typical in cryptanalysis

Additional Results

Solving the RSA partial factorization problem

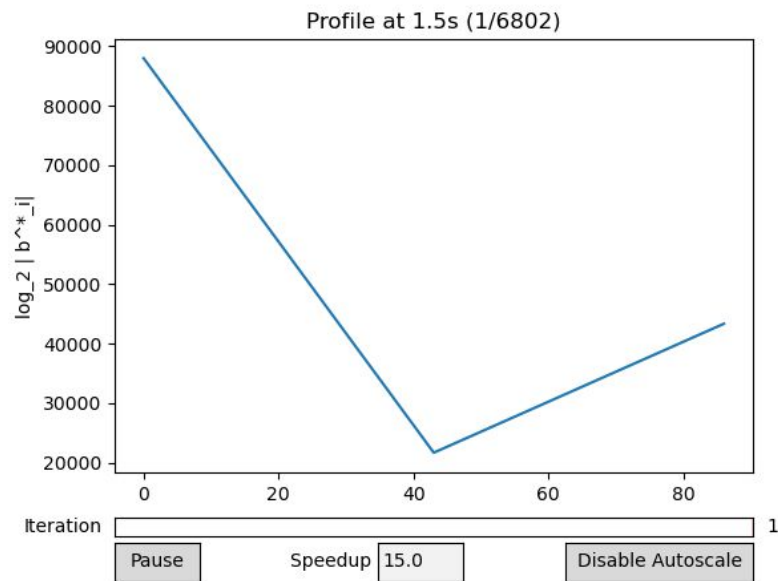
Given a 2048-bit RSA modulus $N = pq$ and 512 most significant bits of p , factor N .

Previous results*:

470 core-hours

Our results:

~18 core-hours



*from [AHMP23], <https://eprint.iacr.org/2023/329>

Solving the Gentry-Halevi FHE problem

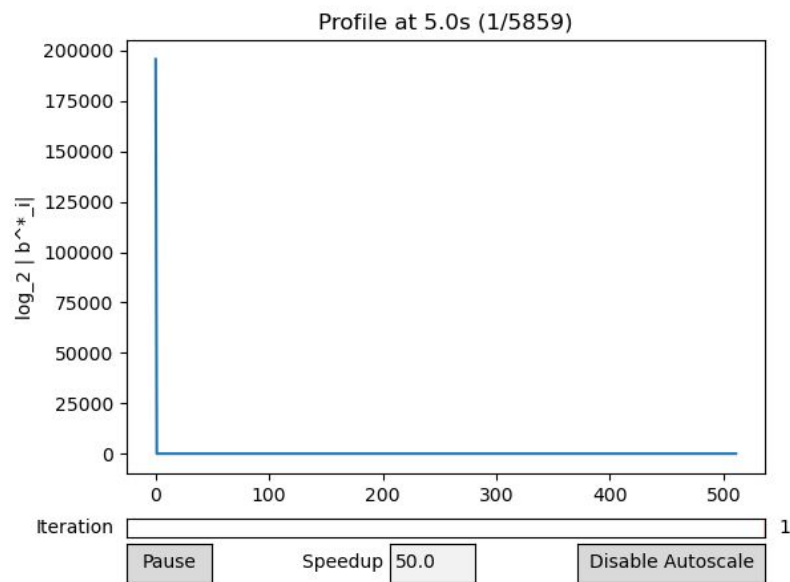
Given a public key for the Gentry-Halevi FHE scheme, recover the private key.

Previous results (toy/small)*:

24 core-days/15.7 core-years

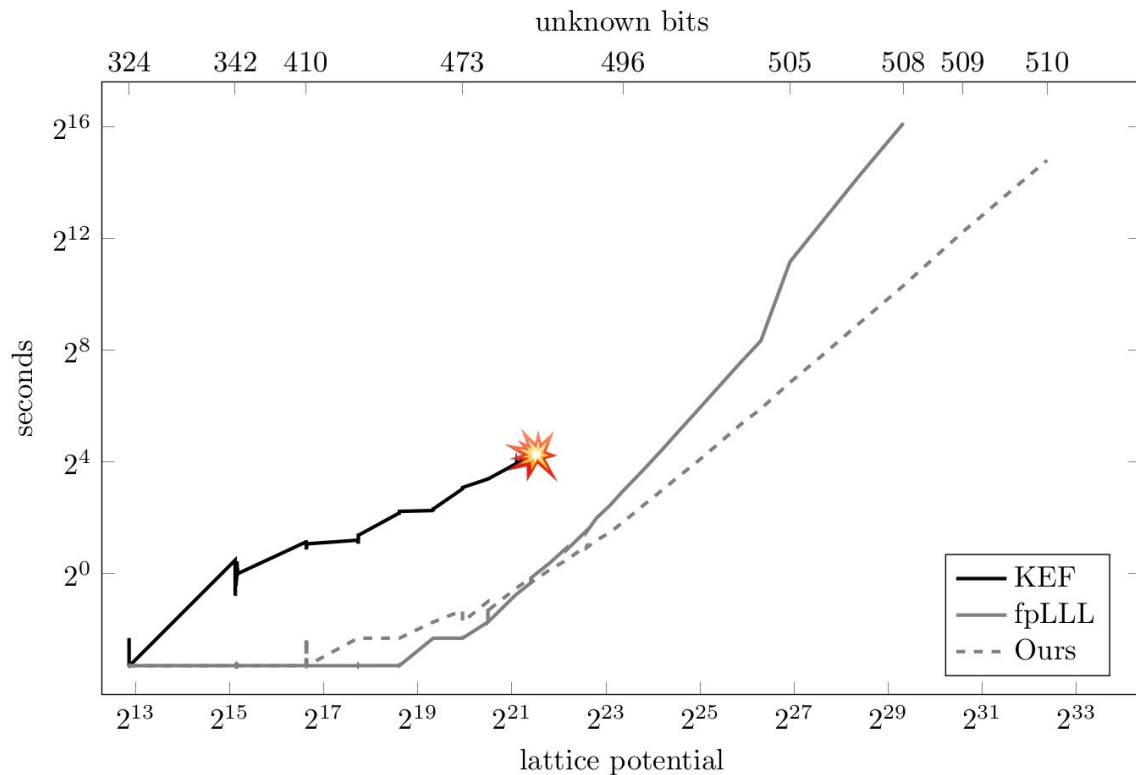
Our results:

15 core-minutes/31 core-hours

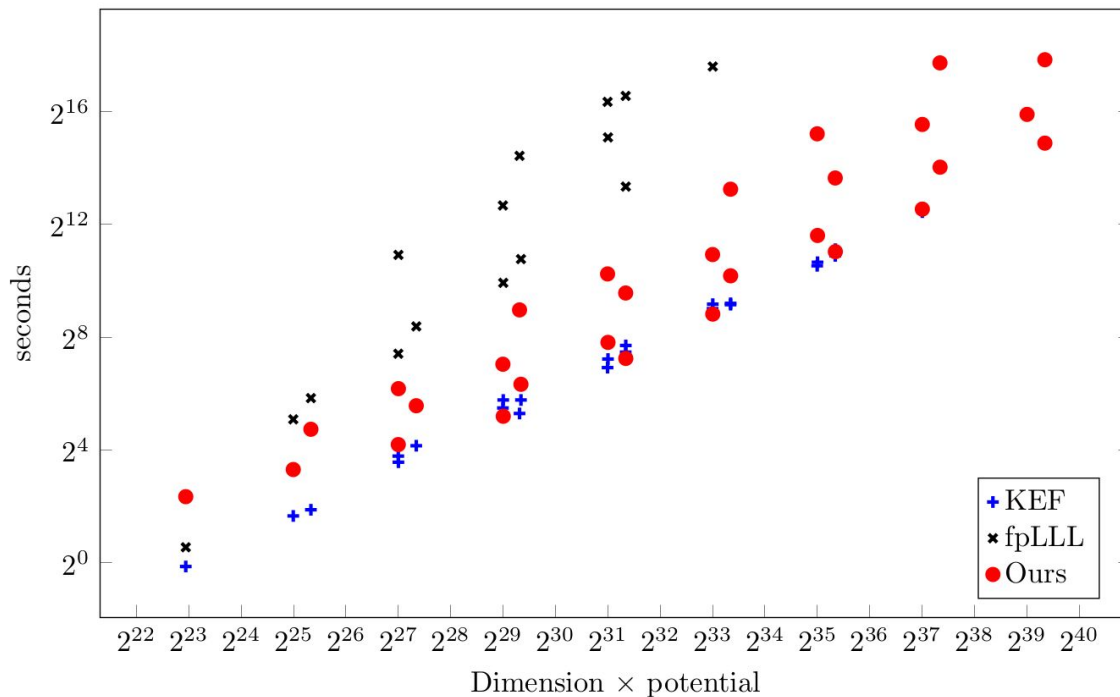


*from [PSZ15], <https://doi.org/10.1007/s10623-014-9957-1>

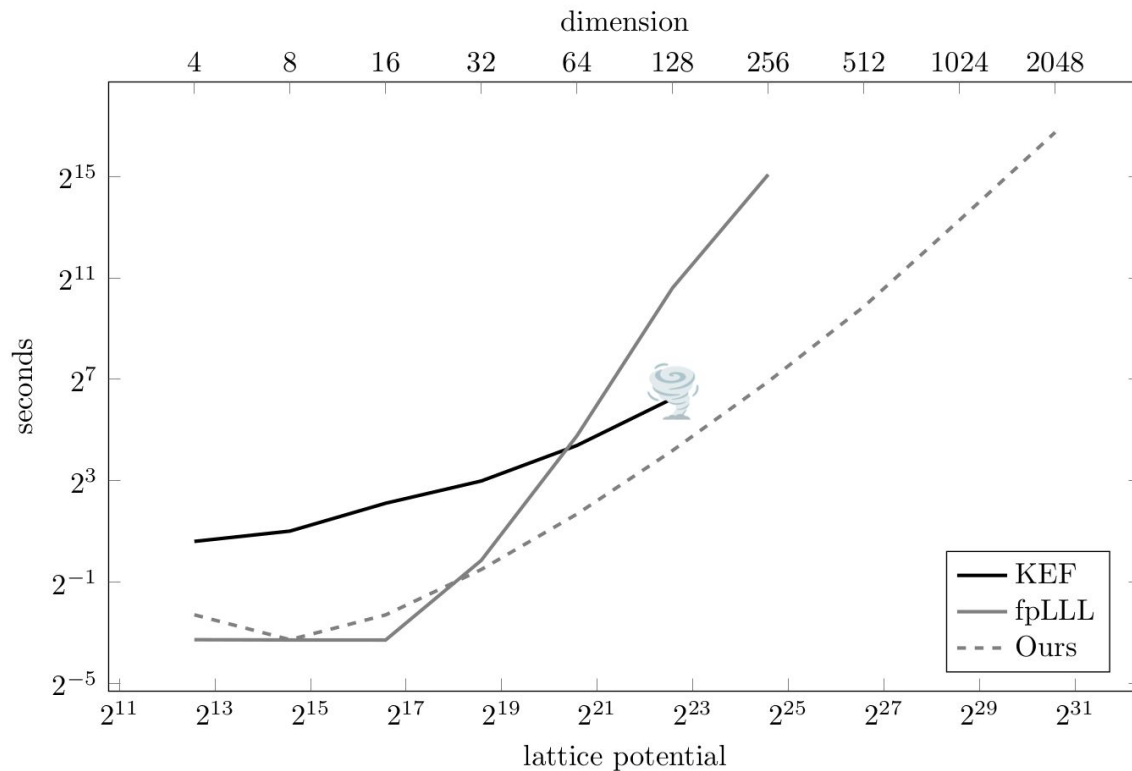
Additional Results - RSA Partial Factorization



Additional Results - Random q -ary bases



Additional Results - Gentry-Halevi FHE



Summary

- Lattice reduction is an important tool for cryptanalysis
- Algorithms can be improved via recursion or precision management
- [KEF21] found a way to combine both, but it doesn't work for all lattices.
- We use the drop instead of the spread to keep the precision small.
- Our code is significantly faster than state-of-the-art implementations.
- Future work may include making our algorithm rigorous or decreasing the running time even further.

Questions?



ia.cr/2023/237



<https://github.com/keeganryan/flutter>

References

- [AHMP23] Albrecht, M., Haller, M., Mareková, L., Paterson, K.: Caveat implementor! Key recovery attacks on MEGA. Cryptology ePrint Archive, Report 2023/329 (2023), <https://eprint.iacr.org/2023/329>
- [KEF21] Kirchner, P., Espitau, T., Fouque, P.A.: Towards faster polynomial-time lattice reduction. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part II. LNCS, vol. 12826, pp. 760790. Springer, Heidelberg, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84245-1_26
- [KS01] Koy, H., Schnorr, C.P.: Segment LLL-reduction of lattice bases. In: Silverman, J.H. (ed.) Cryptography and Lattices. pp. 6780. Springer, Heidelberg (2001)
- [LLL82] Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. *Mathematische Annalen* 261(4), 515534 (Dec 1982). <https://doi.org/10.1007/BF01457454>
- [NS16] Neumaier, A., Stehlé, D.: Faster LLL-type reduction of lattice bases. In: Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation. p. 373380. ISSAC '16, Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2930889.2930917>
- [NS09] Nguyen, P.Q., Stehlé, D.: An LLL algorithm with quadratic complexity. *SIAM Journal on Computing* 39(3), 874903 (2009). <https://doi.org/10.1137/070705702>
- [PSZ15] Plantard, T., Susilo, W., Zhang, Z.: LLL for ideal lattices: re-evaluation of the security of Gentry-Halevi's FHE scheme. *Designs, Codes and Cryptography* 76(2), 325344 (Aug 2015). <https://doi.org/10.1007/s10623-014-9957-1>
- [SMSV14] Saruchi, Morel, I., Stehlé, D., Villard, G.: LLL reducing with the most significant bits. In: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation. p. 367374. ISSAC '14, Association for Computing Machinery, New York, NY, USA (2014). <https://doi.org/10.1145/2608628.2608645>