Theory and Application of Lattice Reduction by Phong Q. Nguyen (阮风光)

Rick

2020 年 10 月 23 日

1 Theory and Application of Lattice Reduction

Phong's talk on Lattice problem reduction is very well-organized, specifically, he gave a *three-line proof* of the $\mathsf{GapSVP} \leq \mathsf{SIS}$ reduction, which is very clear and relatively easy to follow.

At the beginning of the talk, he mentioned that implementing an algorithm helps the understanding (so I considered implementing the LLL and nearest hyperplane with randomized rounding algorithm after reading [GPV08]). He also mentioned two mathematical reference books about the foundations in lattice-based cryptography.

- Lectures on the geometry of numbers
- Geometry of Numbers

2 Ubiquity of Lattice

Lattice has rich application in mathematics, physics, etc.

Phong starts the presentation by introducing that only Euclidean lattice is used in cryptography, moreover, cryptographic schemes implicitly works on the finite abelian quotient group $\mathbb{Z}_n/L$ rather than the infinite L. One notation he used is the so-called $\mathbb{Z}$-span, which is the integer combination of a series of vectors. Let

$$\begin{aligned}
b_1, \ldots, b_n &\in \mathbb{R}^m \\
L(b_1, \ldots, b_n) &:= \{z_1 \cdot b_1 + \ldots + z_n \cdot b_n : z_1, \ldots, z_n \in \mathbb{Z}\}
\end{aligned}$$

There are three Lattice invariants:

- Rank;
- The volume of the fundamental "parallelpiped"

$$\begin{aligned}
\mathrm{vol}(L) &:= \mathrm{vol}(b_1, \ldots, b_n) \\
&= \sqrt{|\det(\langle b_i, b_j \rangle)|} \\
&= \underbrace{\mu}_{\text{tolerance}} (\mathrm{span}(L)/L)
\end{aligned}$$

where the last equation I have no idea what it means;

- The first successive minima, $\lambda_1(L)$.

Then there is duality on lattice. For a lattice L, we can define its dual lattice $L^{\perp} := \{z : z^t \cdot x \in \mathbb{Z}, \forall x \in l\}$, the basis for $l^{\perp}$ is b^{-1^t}. the invariants are also connected:

- $\text{rank}(L) = \text{rank}\left(L^{\perp}\right)$
- $\text{vol}(L) \cdot \text{vol}(L^{\perp}) = 1$
- I am sure other invariants are also connected, but I missed them in the lecture.

The lattice problem is to find short & near orthogonal basis for a given lattice. Lattice reduction (whose purpose is to find such nice basis) initially being proposed in the mathematical world to find invariant connections, there are now more sophisticated (actually he said "much better") methods to build invariant connections (which cryptography has not utilized yet, and this in fact is one of the open problems he mentioned at the end of the talk, if I recall correctly).

In linear algebra, there are orthogonal basis, but that is not the case in Lattice. Nevertheless, reduced basis is the approximation of that notion who are small and nearly orthogonal.

3 Summary

- background
- lattice problems
- worst-case to average-case reductions
- the LLL algorithm
- enumeration
- blockwise algorithm
- security estimates issues

4 Background

This talk is about the mathematical aspects of lattice.

Notebooks

Lectures on the geometry of numbers sigel's book

Geometry of Numbers This book is written by Fields medel wonner Venkatesh

4.1 Euclidean Lattice

Phong expands the notion of lattice by integer lattice which was discussed in the previous lectures to Euclidean lattice.

Consider $\mathbb{R}^n$ as a Euclidean space, a Lattice is a discrete subspace over that Euclidean space. The dimension of a lattice is defined as the dimension of the subspace spannned by this lattice.

$$\dim L = \operatorname{rank}(L) = \dim \operatorname{span}(L)$$

E.g. $\mathbb{Z}^n$ and its subgroups

4.2 Integer Lattice

Interger lattice is just a special example of Euclidean lattice, the only restriction here is that the quotion should be a finite group. Id est, for any discrete subgroup L of $\mathbb{Z}^n$, $\mathbb{Z}^n/L$ should be finite. This is the type of lattice that we work with in crypto.

A lattice is infinite, but in crypto, we work in the finite abelian group $\mathbb{Z}^n/L$, rather than the lattice itself. And that group is what he called we replace RSA or elliptic cruve group with. Every operation is modulo the lattice

1.

And that is way in many lattice crypto, we do not see the lattice directly. We work modulo the lattice.

Here Phong uses a notation $\mathbb{Z}$-span to denote the integer combination of a set of vectors. The notation is $L = L(b_1,\ldots,b_n)$.

4.3 Characterization of Lattice

Let $L \subset \mathbb{R}^m$ be non-empty, then the following two conditions are equivalent.

- L is a lattice
- $\exists b_1,\ldots,b_m \in \mathbb{R}^m$ linearly independent such that $L = L(b_1,\ldots,b_n)$

4.4 Lattice Invariants

Some properties of the lattice is invariant of the choice of basis, and some of them presented are:

- Rank;
- The volume of the fundamental "parallelpiped"

$$\begin{aligned} \mathrm{vol}(L) &:= \mathrm{vol}(b_1, \ldots, b_n) \\ &= \sqrt{|\det(\langle b_i, b_j \rangle)|} \\ &= \underbrace{\mu}_{\text{measure}} \underbrace{(\mathrm{span}(L)/L)}_{\text{torus}} \end{aligned}$$

  In mathematics the volume we defined here is actually called "co-volume", and it corresponds to the measure of the quotient of the subspace spanned by the lattice (span(L)) divided by the lattice itself. The resulting quotient is called a (compact) torus and that is something I have never learned or heard before.

- The first successive minima, $\lambda_1(L)$. Actually any minima is invariant, but in this lecture we are only concerned with the first minimum.

4.5 Duality

It is a key idea in lattice.

Let L be a lattice, its dual lattice is defined as

$$L^\perp := \left\{ z : z^t \cdot x \in \mathbb{Z}, \forall x \in L \right\}$$

This set is a subset of linear space $\mathbb{R}^n$ and itself is discrete. And therefore, a lattice.

A classical brunch of mathematics tries to connect the invariants between a primal lattice and its dual lattice. some of the trivial examples are as follows.

- $\mathrm{rank}(L) = \mathrm{rank}\left(L^\perp\right)$
- $\mathrm{vol}(L) \cdot \mathrm{vol}(L^\perp) = 1$

This comes directely from the the basis for $L^\perp$ being b^{-1^t}.

The connection, say λ_1 is not so trivial.

4.6 Lattice Reduction

In linear algebra, there are orthogonal basis, but that is generally not the case in Lattice. Nevertheless, reduced basis is the approximation of that notion who are small and nearly orthogonal.

4.6.1 Motivations

The lattice problem is to find short & near orthogonal basis for a given lattice. Lattice reduction (whose purpose is to find such nice basis) initially being proposed in the mathematical world to find invariant connections, but there are now much better inequalities, that you can forget about lattice reductions. Mathematicans have developed much more sophisticated tools to get rid of lattice reduction.

But lattice reduction is still very useful to algorithms, and we are kind of stuck with it currently, and this in fact is one of the open problems he mentioned at the end of the talk, if I recall correctly).

5 Mathematical Problems

Basic Question:

- given a subspace C and a lattice L, is $L \cap C$ empty?
- how many lattice points are there in $L \cap C$.

5.1 Minkowski's theorem

Minkowski's theorem states that for a convex symmetrical subset C, if $\mathrm{vol}(C) > 2^n \mathrm{vol}(L)$ then there must be some non-zero lattice point in C. The proof is very simple. Consider the scaled subset $1/2C$. We shift the subset by all the lattice points, and since $vol(1/2C) > vol(L)$, there must be some overlaping vectors. Take one such vector $1/2v$, this means that for some vector $1/2w$, there exists some non-zero lattice point a such that

$$1/2w + a = 1/2v$$

They by the symmetry and contex property, we have

$$a = 1/2v - 1/2w \in C$$

which completes the proof.

5.2 Hermite's constant (1850)

$$\gamma_d := \sup_L \frac{\lambda_1(L)^2}{\mathrm{vol}(L)^{2/d}}$$

Where the square root on γ_d side is for historical reason, as Phong mentioned.

Initially, Hermite used lattice reduction and showed that

$$\gamma_d \leq (4/3)^{(d-1)/2}$$

But by using Minkowski's inequality, one can get a much better bound

$$\gamma_d \leq d,$$

which gives rise to the existence of non-zero lattice vector of norm less or equal to

$$\mathrm{vol}(L)^{1/d} \cdot \sqrt{d}$$

in a d dimensional lattice.

But although the latter inequality is much better, it is not algorithmically efficient to find such vector, so lattice reduction still makes sense.

5.3 The Gaussian Heuristic

For "nice" full-rank lattice and "nice" measureable set C of $\mathbb{R}^n$, we have

$$\mathrm{Card}(C \cap L) \approx \frac{\mathrm{vol}(C)}{\mathrm{vol}(L)}$$

And from that Heuristic, there are two theorems. It is like a phase transition in the length of the radius of the ball.

5.3.1 Theorem 1

Any d-dim lattice L has exponentially many vectors of norm less or equal to

$$O(\sqrt{d})\mathrm{vol}(L)^{1/d}$$

This can be understood as follows. Consider a ball, if we increase its radius, its volume will increase exponentially. By Gaussian heuristic, this means the number of points inside the ball also grows exponentially. Notice that this is a mere explanation, **not** a proof.

5.3.2 Theorem 2

In a random d-dim lattice L, all non-zero lattice vectors have norm greater than or equal to

$$\Omega(\sqrt{d})\text{vol}(L)^{1/d}$$

The intuition behind this theorem is that if we decrease the aforementioned ball's radius, the number of inside lattice points will decrease. This result is true to **random** lattice.

6 Lattice Problems

Since in crypto we are only interested in integer lattice, it is common to consider input as a matrix whose rows are lattice basis. Two parameter here are of interest:

- Dimension of the lattice
- Size of the coefficients

We normally consider the size of the coefficients polynomial in the dimension.

6.1 Hard lattice problems

Since 1996, lattices are very trendy in classical and quantum complexity theory. Below is a table of the hardness of what I presume GapSVP_γ problem.

Hardness	Approximation Factor
NP-hardness	$O(1)$
non NP-hard NP ∩ coNP	$\sqrt{d}$
worst-case/average-case reduction	$d\log d$
cryptography	$d^{O(1)}$
subexp-time algorithms	$2^{\sqrt{d}}$
poly-time algorithms	$2^{\frac{d\log\log d}{\log d}}$

The question here is that is crypto hard enough?

6.2 Hard lattice problem in cryptography

The general format of such problem is Input: a lattice L and a n-dim ball C Output: Decide if $L \cap C$ is non-trivial and find a point when applicable

There are two settings

Approximate $L \cap C$ has many points, e.g. SIS and ISIS

Unique there is only one unique point, e.g. BDD

6.3 The Shortest Vector Problem (SVP)

Input: a basis of a d-dim lattice L Output: a non-zero v such that $\|v\| = \lambda_1(L)$

And its relaxed version, since this problem is proved to be NP-hard under randomized reduction. There are two flavors, but they are equivalent to some extent.

Approximate-SVP find vector v such that $\|v\| \leq f(d) \cdot \lambda_1(L)$

Hermite-SVP find v such that $\|v\| \leq g(d) \cdot \mathrm{vol}(L)^{1/d}$

The difference is that solutions to the first problem is hard to check, since the first minima is not easily computable.

6.4 The Closest Vector Problem (CVP)

Input: a basis of a lattice L of dim d, and a target vector t Output: a vector v minimizing $\|v - t\|$

BDD problem is closely related, but with the constraint that the target vector is rather close to some lattice point.

6.5 Random Instances

How to generate a lattice for the previous hard problems? how to choose the noise and the secret lattice vector?

More specifically, how to prove that the certain way used to generate lattice instance does not bring advantage to solving this problem, compared with other distributions?

Mathmaticians have already studied this problem in the real lattice. There is a “natual” notion of **random lattice** due to [Siegel45] and its related to Haar measures. And in fact, this is one of the “advanced tools” he mentioned earlier.

Also there are examples of random lattice properties that are known. E.g. [Rogers56] points out the limit distribution of the volume of the ball of radius $\lambda_1(L)$ when d approaches infinity, is an exponential distribution.

But that are all real lattices, rather than integer lattice.

Note that a full-rank integer lattice $L \subset \mathbb{Z}^m$ defines a finite Abelian group $\mathbb{Z}^m/L$ of rank less than or equal to m.

Reciprocally, for any finite abelian group G, we can define

$$L_m(G) := \{L \subseteq \mathbb{Z}^m : \mathbb{Z}^m/L \sim G\}$$

which is finite. Moreover, they define a partitiion of all the full rank lattices, when G ranges over all the finite abelian groups.

A very powerful theorem from ergodic theory kind of solves this problem of choosing "proper" random integer lattice. The theorem is as follows: [EsOh04]: If L_n is chosen uniformly at random from $L_m(G_n)$ where (G_n) is a sequence of finite abelian groups of order towards infinity, then the distribution of L_n "converges" to the Haar distribution.

Note that here the only requirement of abelian groups are that their order grows to infinity.

6.6 The SIS problem

And now comes the SIS problem. Mostly SIS is defined with $G = \mathbb{Z}_q^n$, but actually any finite abelian group suffices.

Take any finite abelian group, and view it as a Z-module. (It means that 2G = G+G) Pick $g_1, \ldots, g_m$ uniformly at random from G, and then find short $(x_1, \ldots, x_m) \in \mathbb{Z}^m$ such that $\sum_i x_i g_i = 0$.

This essentially is finding a short vector in a uniform random lattice of

$$L_m(G) = \{L \subseteq \mathbb{Z}^m : \mathbb{Z}^m/L \sim G\}$$

The way I understand that typical example (G = $\mathbb{Z}_p^n$) is as follows. Consider a homomorphism σ

$$\sigma : \mathbb{Z}^m \to \mathbb{Z}_p^n$$
$$\sigma : v \mapsto A \cdot v$$

for some full rank $A \in \mathbb{Z}_p^{n \times m}$ and $m > n$. Now we can define a isomorphism σ':

$$\sigma' : \mathbb{Z}^m/\ker(A) \to \mathbb{Z}_p^n$$

Here the operation is still left multiply by A, but now its an isomorphism. By using different such A, we can then acquire differnet kernels, and finally different quotient groups isomorphic to $G = \mathbb{Z}_p^n$, which is exactly the definition of $L_m(G)$. So now the vanilla SIS problem can be translated into group theoretic language by starting with a series of finite abelian groups

$G=\mathbb{Z}_p^n$ that fits the aforementioned requirement (order goes to infinity), and then randomly picking vectors $g_1,\dots,g_m$ that essentially defines an integer lattice $L=\mathrm{ker}(A)$, having the effect of randomly picking a lattice in the finite set $L_m(G)$, and finally find a short vector inside this lattice.

Now we use a different group, say $G=\mathbb{Z}/q\mathbb{Z}$, and randomly sample $g_1,\dots,g_m \leftarrow \$ G$ and seeks to find $x_1,\dots,x_m\in\mathbb{Z}^m$ such that $\sum_i x_i\cdot g_i=0(\mod q)$

And that is finding a short vector for random lattice L such that $\mathbb{Z}^m/L\sim\mathbb{Z}/q\mathbb{Z}$

7 worst-case to average-case reductions

In [Ajtai96], the author proved that if one can efficiently solve SIS for $G=(\mathbb{Z}/q_n\mathbb{Z})^n$ on average, then one can efficiently find short vectors in every n-dim lattice L.

Notice here the modulus q depends on the dimension n, and by the previous "ergodic" theory results, so long as the order of this finite abelian group grows to infinity, we should not be constrained with $\mathbb{Z}_q^n$.

So [GINX16] showed that the result can be generalized to any sequence G_n of finite abelian groups, provided that

$$\#G\geq n^{\Omega(\max(n,\mathrm{rank}(G)))}$$

and m is too sufficiently large and growing fast.

7.1 The framework of this reduction

Since we are doing a worst-case to average-case reduction, there must not be any assumption on this lattice L. Now define a super lattice $\bar{L}=q^{-1}L$ so that $\bar{L}/L=(\mathbb{Z}/q\mathbb{Z})^n$, and we denote $G=(\mathbb{Z}/q\mathbb{Z})^n$.

Then there is an **exact sequence**:

$$0\to L\xrightarrow{1}\bar{L}\xrightarrow{\phi}G\to 0$$

And $L=\ker\phi$, ϕ is efficiently computable.

Let $v_1,\dots,v_m\in\bar{L}$ and define $g_1,\dots,g_m\in G$ by $g_i=\phi(v_i)$.

If $\sum_i x_ig_i=0$ for $x_1,\dots,x_m\in\mathbb{Z}^m$, then $\sum_i x_iv_i\in L$.

So now we have a map from vector in the over lattice into lattice vectors. If we somehow find a way so that g_i are uniform and v_i are small, then the SIS Oracle will return a valid solution (since we are querying in the average case), and then the resulting vector $\sum_i x_i v_i \in L$ is also small.

The only question now is how to find such distribution that enjoys this property.

The answer is **harmonic analysis**. If a distribition over $\bar{L}$ satisfies that for any $x \in \bar{L}, \sum_{y\in L} f(x+y) \approx 1/\#G$, where f is the PDF, then the mapped $g_i = \phi(v_i)$ has uniform distribution over G.

This step allows us to use one lattice-invariant method to solve all the lattice problem. Needless to say, this is the key step.

7.2 The 3-line reduction

The worst-case to average-case SIS reduction in three lines:

1. Sample short random vectors $v_1, \ldots, v_m \in \bar{L}$, so that $g_i = \phi(v_i)$ has unifrom distribution over G.

2. Call the SIS oracle on $g_1, \ldots, g_m$ to find a short solution $x_1, \ldots, x_m$ such that $\sum_i x_i g_i = 0$ and so $\sum_i x_i v_i \in L$.

3. Return $\sum_i x_i v_i \in L$

7.3 Generalized SIS Reduction

Notice that the way we create the overlattice $\bar{L}$ preserves the reduced basis from L, which is essential in sampling short vectors in the overlattice. But this might not be the case for general G.

In the general case, one need to find a **reduced basis** of some over lattice $\bar{L}$ such that $\bar{L}/L = G$, so that we can sample some short vector in $\bar{L}$.

7.4 Closest Vector Group

Previously we have discussed the shortest vector problem. But how about the closest vector problem? (How to we choose the target so that the hardness of this problem is not affected by our particular choice).

Remember the SIS lattice:

- $g_1, \ldots, g_m$ for some finite Abelian group $(G, +)$

- define the lattice $L = \{x = (x_1, \ldots, x_m) \in \mathbb{Z}^m : \sum x_i g_i = 0\}$

The dual lattice of L is related to the dual group G^x of (additive) characters of G: morphisms from G to torus $T = \mathbb{R}/\mathbb{Z}$.

Then define dual lattice

$$L^x = \{(y_1, \ldots, y_m) \in \mathbb{R}^m : \exists s \in G^x, \forall i, y_i = s(g_i) (\mod 1)\}$$

7.5 The LWE Problem

The aforementioned dual lattice gives rise to another view of the LWE problem. Let G be any finite Abelian group, e.g. $G = (\mathbb{Z}/q\mathbb{Z})^n$, and pick $g_1, \ldots, g_m \leftarrow \$ G$ uniformly at random. Then pick a random character s in G^x.

The goal is to recover s given $g_1, \ldots, g_m$ and noisy approximation of $s(g_i)$.

The natural distribution of noise inside the torus is Gaussian distribution. And the property of Gaussian noise is that when σ gets too large, the distribution inside the torus ges too close to uniform – corrupting the character completely.

Example: cyclic G Let $G = \mathbb{Z}/q\mathbb{Z}$, and pick $g_1, \ldots, g_m \leftarrow \$ G$. The goal is to recover $s \in \mathbb{Z}$ given $g_1, \ldots, g_m$ and randomized approximations of $sg_1, \ldots, sg_m \mod q$.

This is exactly a randomized variant of Boneh-Venkatesan's Hidden Number Problem of Crypto'96.

[Regev05] showed that if one can efficiently solve LWE for $G = (\mathbb{Z}/q\mathbb{Z})^n$ on the average, then on can quantum-efficiently find short vectors in every n-dim lattice.

Also in [GINX16], the author shows that this can be generalized to any sequence (G_n) of finite abelian groups provided the order $\#G_n$ is sufficiently large.

7.6 Ring Variants

In order to achieve concrete efficiency, we can use special structure inside lattice for better efficiency, at a cost of stronger hardness assumption. This is the case with NTRU [HPS98].

Starting with [Mi02], one can obtain "restricted" worst-case to average case reduction in the sense that the worst case now refers to the that in a special class of lattices, for instance (but not limited to) ideal lattices.

This is a paractical tradeoff.

This restricted reduction is then further explained.

Let M be a finite R-module for some ring R. $R = \mathbb{Z}$ in SIS. Pick $g_1, \dots, g_m \leftarrow\$ M$ uniformly at random. Since M is finite, we can do this. The goal is to find $(x_1, \dots, x_m) \in R^m$ such that $\sum_i x_i g_i = 0$ (in the module).

If R^m is a lattice, then this is finding a short vector in some random (module) sublattice of R^m.

7.6.1 Example

One example is NTRU. There

$$\begin{aligned} m &= 2 \\ R &= \mathbb{Z}[X]/(X^N - 1) \\ M &= \mathbb{Z}[X]/(q, X^N - 1) \end{aligned}$$

But $g_1 = \mathsf{pk}, g_2 = -1$, instead of uniformly random.

7.6.2 Reduction

[LaSt14]: If one can efficiently solve M-SIS for $M = (R/qR)^d$ where R is the ring of integers of a cyclotomic field, then one can efficiently find short vectors in every module lattice of R^d.

It is a generalization of the previous ideal-lattice reductions for $d = 1$.

There are similar results for M-LWE [LaSt14] generalizing Ring-LWE's hardness [LPR10].

8 the LLL algorithm

In mathematics a classical problem is to prove the **existence** of short lattice vectors.

In the past thirty to forty years, crypto has been rediscovering mathematical statements on Hermite's constant bounds, Phong commented. In particular, all known upper bounds on Hermite's constant have an algorithmic analogue:

Hermite's Inequality the LLL algorithm

Mordell's Inequality Blockwise generalization of LLL

Mordell's proof of Minkowski's inquality worst-case to average-case reductions for SIS and sieve algorithms [BJN14, ADRS15].

8.1 SVP Algorithms

- Poly time approximation algorithms
    - LLL algorithm [LLL82]
    - Block generalization by [Schnorr97,GHKN06,GamaNG...,MiWa16,ALNSD19]
- Exponential exact algorithms
    - Poly-space **enumeration** [Pohst81,Kannan83,ScEu94]
    - Exp-sapce **sieving** [AKS01,MV10]

8.2 Basis and Filtration

Why are we stuck with lattice basis reduction, rather than the more advanced mathematical tools developed recently? Phong has these to say:

If $(b_1,\dots,b_d)$ is a basis of L, then define $L_i = L(b_1,\dots,b_i)$, which is a sublattice of L. In abstract algebra, this sequence of sublattices is called a flag of L.

Actually this can be viewed as a filtration of lattices. If $i \le j$, then the quotient L_j/L_i is a **lattice** of rank $j-i$, and $\mathrm{vol}(L_j/L_i) = \mathrm{vol}(L_j)/\mathrm{vol}(L_i)$.

By using these low-rank lattices, one can fine-grain complexity. Id est, one works inside the low-rank derived lattices and somehow lift the result into the high-rank original lattice L.

8.3 Hermite's Inequality and LLL

Recap: Hermite proved in 1850

$$\gamma_d \le \gamma_2^{d-1} = \left(\frac{4}{3}\right)^{(d-1)/2}$$

Intuitively, we should be able to find such short vectors using the result in 2-d lattice.

LLL algorithm finds in poly time a non-zero lattice vector v

$$\|v\| \le (4/3+\varepsilon)^{(d-1)/4}\mathrm{vol}(L)^{1/d}$$

and it can be viewed as an algorithmic version of Hermite's inequality.

8.3.1 KEY IDEA

L/L_1 has rank $d-1$ with $\|b_1\| = \mathrm{vol}(L)/\mathrm{vol}(L/L_1)$. If $b_2 \mod L_1$ satisfies Hermite's inequality in L/L_1 one can modify b_2 such that

$$\|b_2\|^2 \leq \|b_2 \mod L_1\|^2 + 1/4\|b_1\|^2$$

And then if $\|b_1\| \leq \|b_2\|$ the b_1 satisfies Hermite's inequality in L. Otherwise, swap(b_1, b_2) and restart.

LLL: taking a quotient lattice and lifting.

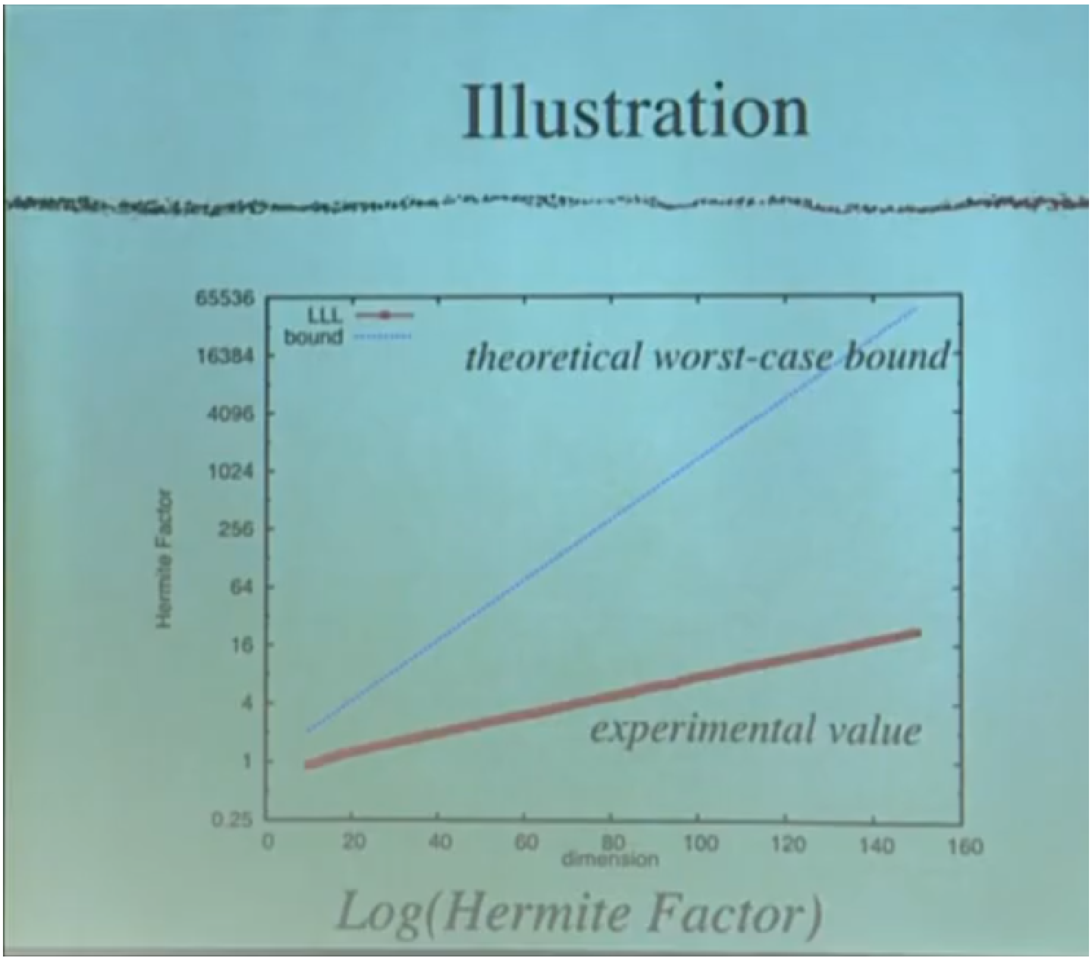
8.4 The Magic of LLL

One of the main reasons behind the popularity is that it performs "much better" than what we the worst-case bound suggest, especially in the low dimension.

This is another example of worst-case vs. "average-case" and the difficulty of security estimates.

Though the approximation factor $(4/3+\varepsilon)^{d-1/4}$ is tight in the worst case and for uniformly ranodm LLL bases [KiVe16]. Experimentally, $4/3+\varepsilon$ can be replace by a smaller constant ≈ 1.08, for any lattice, by randomizing the input basis.

[DKTW19,KiWa19] proposed a **partial explanation** for this gap phenomenon, based on sandpiles.



Illustration

8.5 Open Problems

The gap leaves us with several open problems to ponder upon. Take a random integer lattice L, and let B be the Hermite normal form of L.

- Is it true that with overwhelming probability, after LLL-reducing B,

$\|b_1\| \leq c^{d-1}\mathrm{vol}(L)^{1/d}$ for some $c < (4/3)^{1/4}$?

- Can we guess the distribution of $\|b_1\|$ and the running time?

9 Enumeration

Consider we have for a lattice, a reduced basis, what can we do with it?

Enumeration is the simplest method to solve hard lattice problems, going back to the 1970s.

Input: a lattice L and a small ball $S \subseteq \mathbb{R}^n$ s.t. $(S \cap L)$ is **small**.

Output: All points in $L \cap S$.

We can use enumeration, but there is a severe drawback – running time is typically super-exponential, much larger than $\#(L \cap S)$.

9.1 Enumeration Insight

- Projection never increase norms: if $\|v\| \leq R$, then $\|v \mod L_i\| \leq R$;
- L/L_j is a lower-rank lattice, whose short vectors can be lifted into short vectors of L/L_i if $i < j$ – size reduction.

9.2 Enumeration Steps

1. Reduce a basis
2. Exhausive search all vectors $\leq R$ by enumerating all short vectors in 1-rank L/L_{d-1}, then by lifting we can find all short vectors in 2-rank lattice L/L_{d-2}, and so on.

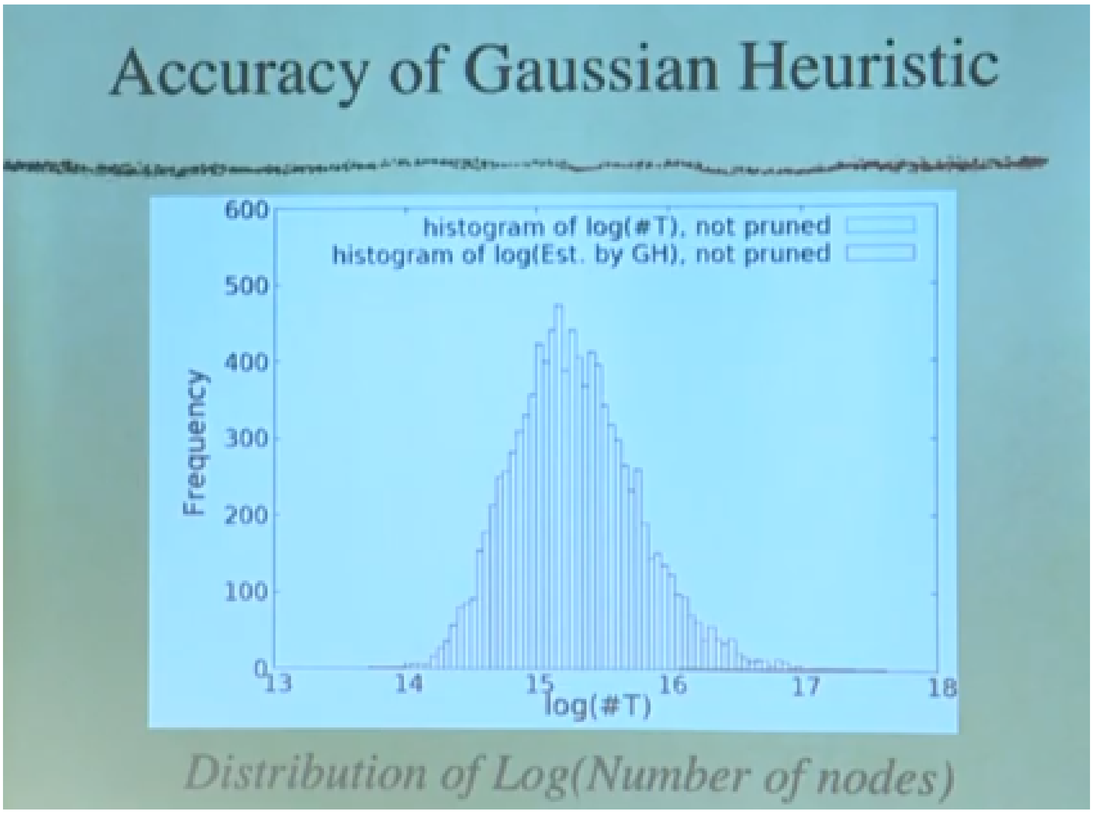
Usually the second step is much more expensive than the first one. If the basis is LLL-reduced, the 2. costs $2^{O(d^2)}$. [Kannan83] showed that 1. and 2. can both be done in $2^{d \ln d}$ poly-time operations.

9.3 Enumeration Complexity

The complexity of enumeration is up to a polynomial factor, the number of lattice points in all quotient lattices inside the ball of radius R. We do that for all the quotient lattices appearing in the algorithm.

Gaussian Heuristics seems very accurate in this case.

Experiment result on the accuracy of Gaussian Heuristics, notice how the two are close enough so that the difference is almost unnoticeable.


Accuracy of Gaussian Heuristic

Distribution of Log(Number of nodes)

9.4 Take Away

- Enumeration is based on one key idea: filtration to decrease the lattice rank.
- Once parameters are fixed, it is possible to reasonably estimate the running time
- Enumeration can be significantly sped up by pruning in practice, which slices a ball in a randomized manner.

10 Blockwise Algorithm

LLL algorithm relies on local reduction in dimension 2. Blockwise algorithm seeks to increase the performance of this algorithm by increasing this number.

Blockwise algorithms find shorter vectors than LLL by using an exact SVP-subroutine in low dim k called blocksize.

Thanks to better enumeration techniques, this subroutine can be done using $2^{O(k)}$ poly-time operations [AKS01,MV10,ADRS15], which is poly in d if $k = \log d$.

10.1 Mathematical Analogy

Mathmatical Analog of this question is “if we show the existence of very short lattice vector in dim k, can we prove the existence of very short lattice vectors in dim d > k?”

[Mordell44]’s inequality generalizes Hermite’s inequality:

$$\sqrt{\gamma_d} \leq \sqrt{\gamma_k}^{(d-1)/(k-1)}$$
$$\lambda_1(L) \leq \sqrt{\gamma_k}^{(d-1)/(k-1)} \mathrm{vol}(L)^{1/d}$$

10.2 Approximation Algorithms for SVP

These can be considered algorithmic analog of upper bounds on Hermite’s constant, i.e. one for existential and one for constructional.

- [LLL82] corresponds to [Hermite1850]’s inequality.

$$\lambda_1(L) \leq \gamma_2^{d-1/2} \cdot \mathrm{vol}(L)^{1/d} = (4/3)^{d-1/2} \cdot \mathrm{vol}(L)^{1/d}$$

- Blockwise algorithms [Schnorr87,GHKN06,GN08,MW16,ALNSD19] are related to [Mordell1944]’s inequality.

$$\lambda_1(L) \leq \sqrt{\gamma_k}^{(d-1)/(k-1)} \mathrm{vol}(L)^{1/d}$$

All known proofs to Mordell’s inquality uses duality. In the proof of Hermite’s inequality, the lattice rank was decreased by considering the quotient L/L_1.

Duality provides another way to reduce the dimension. If L is a d-rank lattice and $v \in L^T$ is non-zero, then $L \cap v^T$ is a (d-1)-rank sublattice.

Here v^T is a hyperplane orthogonal to the dual vector v.

[GamaN2008,MW16,ALNSD19] solves Hermite-SVP with factor essentially

$$\sqrt{\gamma_k}^{(d-1)/(k-1)}$$

using a k-dim SVP-oracle.

They are to Mordell’s inequality what LLL is to Hermite’s inquality: they call the SVP-oracle on L_j/L_i or its dual. (May be the SVP-oracle can be understood as the induction step?)

By choosing an appropriate $k = \log d$, the whole algorithm is poly-time with a subexponential approximation factor.

10.3 Limits of Approximation Algorithms

Since Mordell's inequality can be tight, it seems difficult to improve the block strategy.

If the algorithm also provides an absolute upper bound on the output, it implicitly gives an upper bound on Hermite's constant. E.g. LLL and blockwise algorithms.

11 Security Estimates Issues

Experimentally [MiWa16], not many differences between blockwise algorithms, despite different theoretical bounds. So people are happy with an old algorithm called **BKZ** [ScEu94,CN11] and its widely used in lattice "record" computations.

BKZ tries to decrease cyclically $\mathrm{vol}(L_i/L_{i-1})$ for $i = 1, \ldots, d$ by calling the k-dim SVP subroutine on each quotient $L_{\min d,i+k}/L_{i-1}$.

Then each $\mathrm{vol}(L_i/L_{i-1})$ is locally minimal among certain perturbated bases.

This algorithm still has a gap between the theoretical worst-case bound and the practical performance (as in LLL). But using random lattice theory, one can devise some heuristics that predicts the practical performance.

Recent progress predicts the behaviour of high-blocksize state-of-the-art BKZ ($k \geq 50$), using an efficient simulation algorithm: the minimum of most k-rank L_j/L_i seems to behave like random lattices.

Experimentally, when blocksize increases, the two behaviour seem to converge, but this is still a very strong assumption (these two performs identically when converges.)

11.1 Security Estimates

It is somewhat independent of security proofs.

This is done by identifying the best attack based on the state-of-the-art.

1. Find as many attacks as possible
2. Identify the "best" one
3. Select keysizes / parameters accordingly.

It is somehow hard since cryptanalysis takes time, and the more parameters it takes, the longer it takes. Actually, Phong proposed using machine learning to optimize the parameter choice in those attacks automatically.

[LenstraVerheul00] suggested to

1. model the performance of the best algorithm known, based on record benchmarks and
2. add a security margin by speculating once
   (a) Hardware improvements
   (b) Algorithmic improvements

to decide key size. Of course this is not easy.

What we need in determining parameters, is a lower bound or distribution of the running time.

'we do not have that level of precision' – Phong

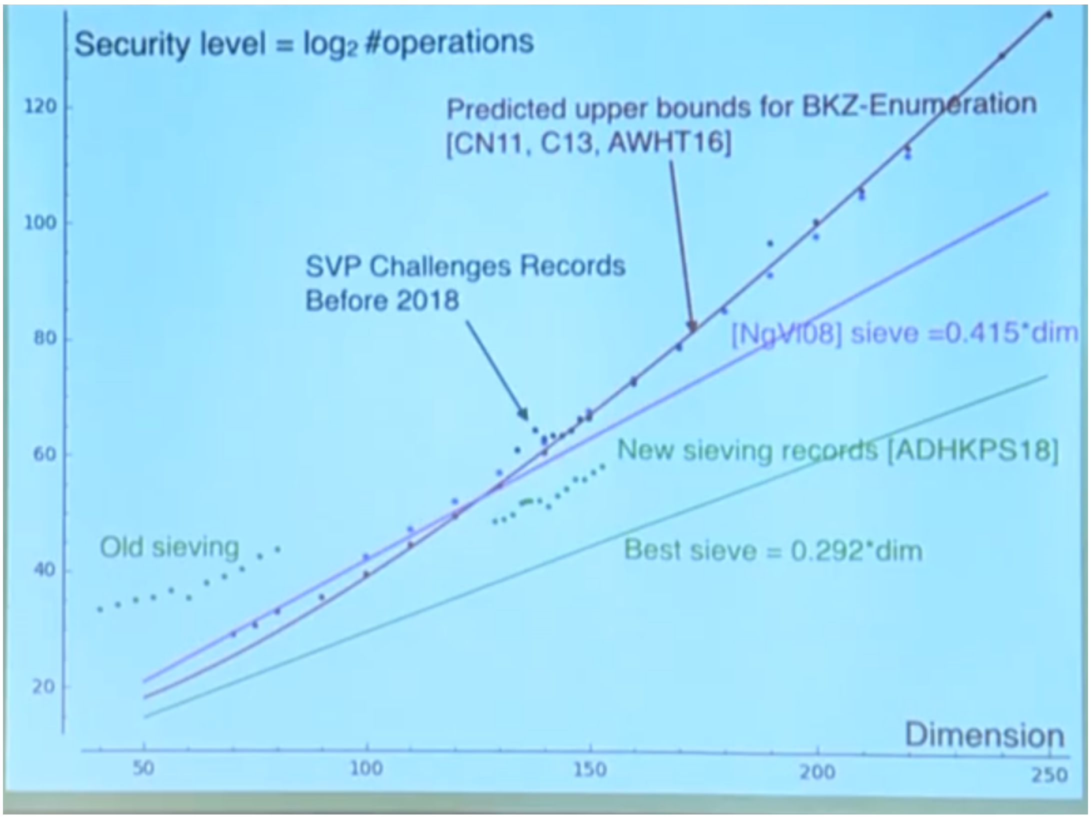
In parctice, lattice-based submission to NIST typically rely on some script to assess security. But existing scripts do not fully reflect various uncertainties of the state-of-the-art. There are so many things that could go wrong.

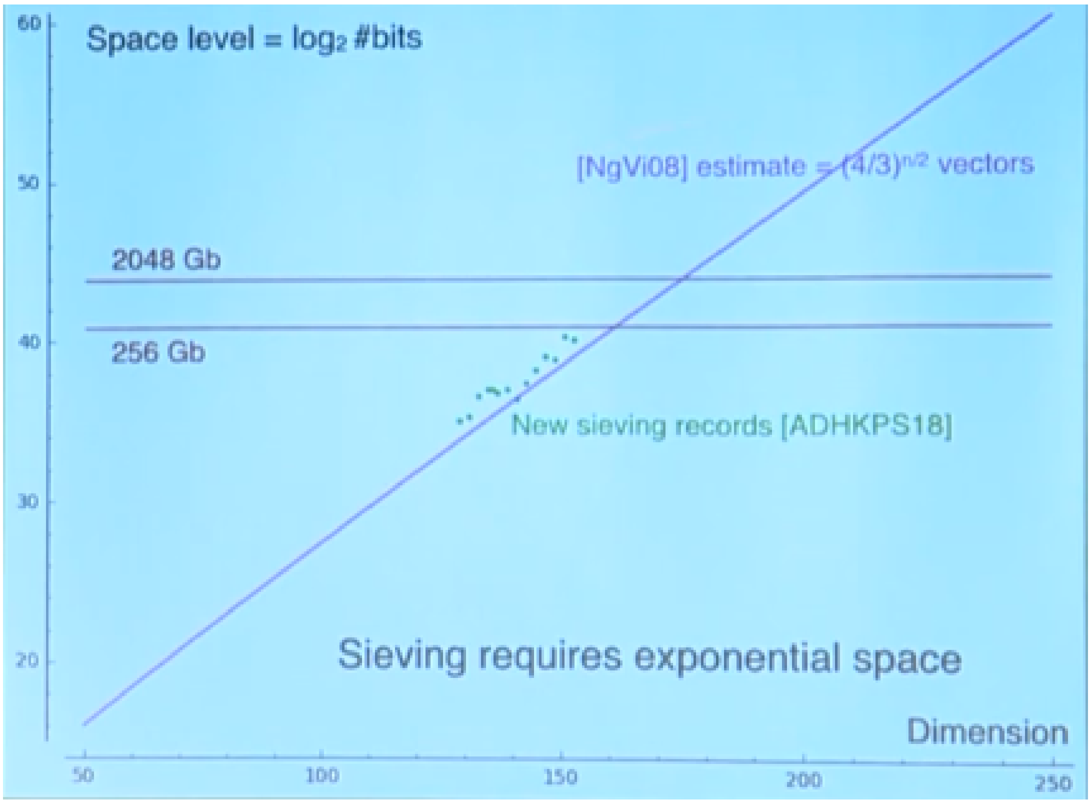
A popular paper estimates the security level on a too precise level that is not accurate.

What is actually done in estimating security parameters is that all proposals claim that the best attack requires to run the SVP subroutine in some target blocksize, so they need to estimate the cost of the subroutine, as well as the number of calls to the subroutine. All being very hard.

Altogether, the current way may gives people a false sense of security.

E.g. SHA-256 in Bitcoin


Security level = log₂ #operations
Predicted upper bounds for BKZ-Enumeration [CN11, C13, AWHT16]
SVP Challenges Records Before 2018
[NgVi08] sieve =0.415*dim
New sieving records [ADHKPS18]
Old sieving
Best sieve = 0.292*dim
Dimension
120
100
80
60
40
20
50
100
150
200
250



It seems from the figure that the space consumption has not progressed much.

12 Conclusion

Lattice Reduction was:

- Introduced to prove the first inequalities on lattice invariants (Hermite's constant)
- Then superseded in mathematics by more sophisticated tools
- But still key in current lattice algorithm

So can we further improve lattice reduction algorithms using these powerful tools?

Can lattice reduction take advantage of special lattices?

Do we really need lattice reduction to solve lattice problems efficiently? (in sieving)

OPEN PROBLEM

An algorithm to approximate SVP within a polynomial factor, possibly quantum or subexponential.