Lattice-based Cryptography

Phong Nguyễn

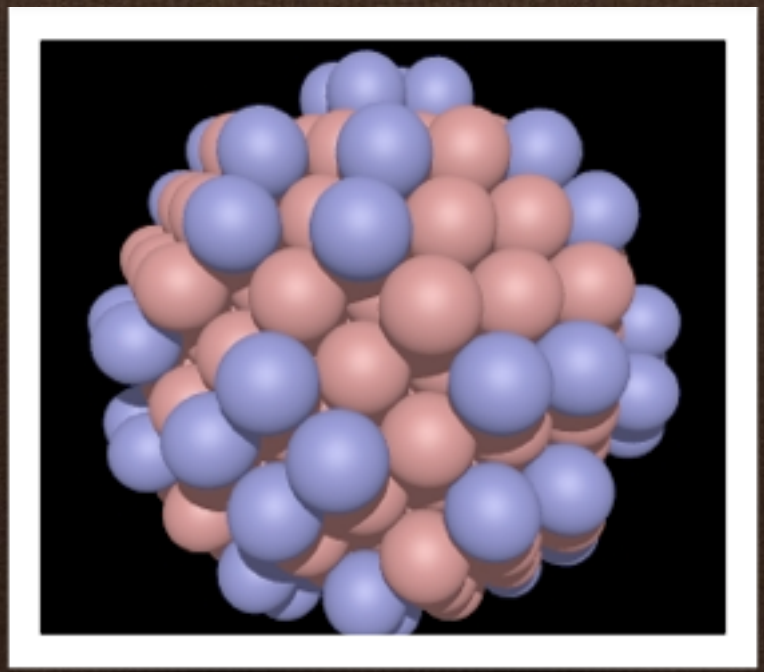
February 2016

Opening

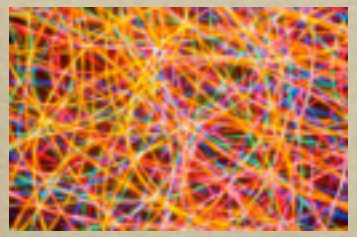
- It is an exciting time, ten years after the 1st Post-Quantum Crypto conference.
- A competition is likely to significantly improve the state-of-the-art.

Summary

- Lattices
- Lattice-based Cryptography
  - Design
  - Cryptanalysis
    - Algorithms
    - Security Estimates



Lattices

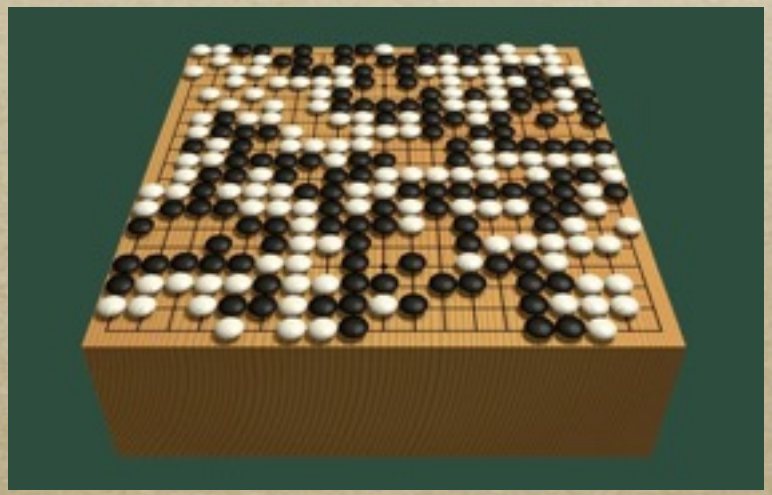


The Ubiquity of Lattices

- In mathematics
  - Algebraic number theory, Algebraic geometry, Sphere packings, etc.
  - Fields medals: G. Margulis (1978), E. Lindenstrauss and S. Smirnov (2010), M. Bhargava (2014).
- Applications in computer science, statistical physics, etc.

What is a Lattice?

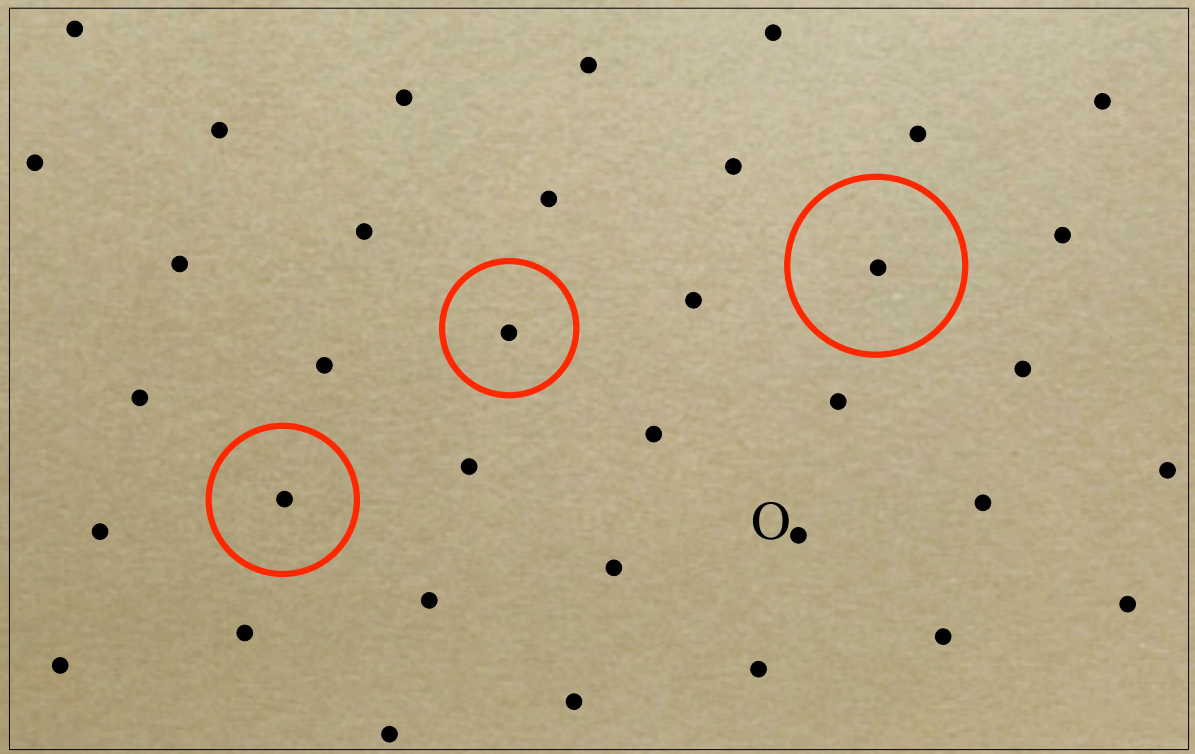
- An infinite arrangement of "regularly spaced" points

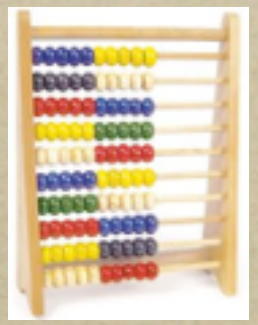


What is a Lattice?

- A lattice is a discrete subgroup of $\mathbf{R}^n$, or the set $L(b_1,\ldots,b_d)$ of all linear combinations $\Sigma x_i b_i$ where $x_i \in \mathbf{Z}$, and the b_i's are linearly independent.

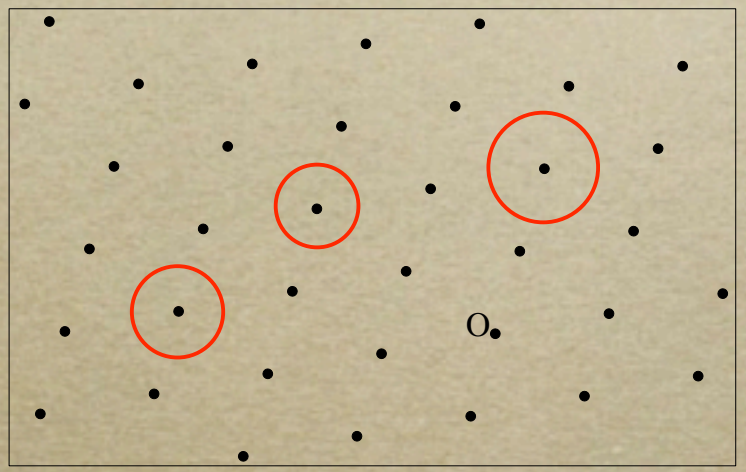
2	0	0	0	0
0	2	0	0	0
0	0	2	0	0
0	0	0	2	0
1	1	1	1	1





Integer Lattices

- A (full-rank) integer lattice is any subgroup L of $(\mathbf{Z}^d,+)$ s.t. $\mathbf{Z}^d/L$ is finite.

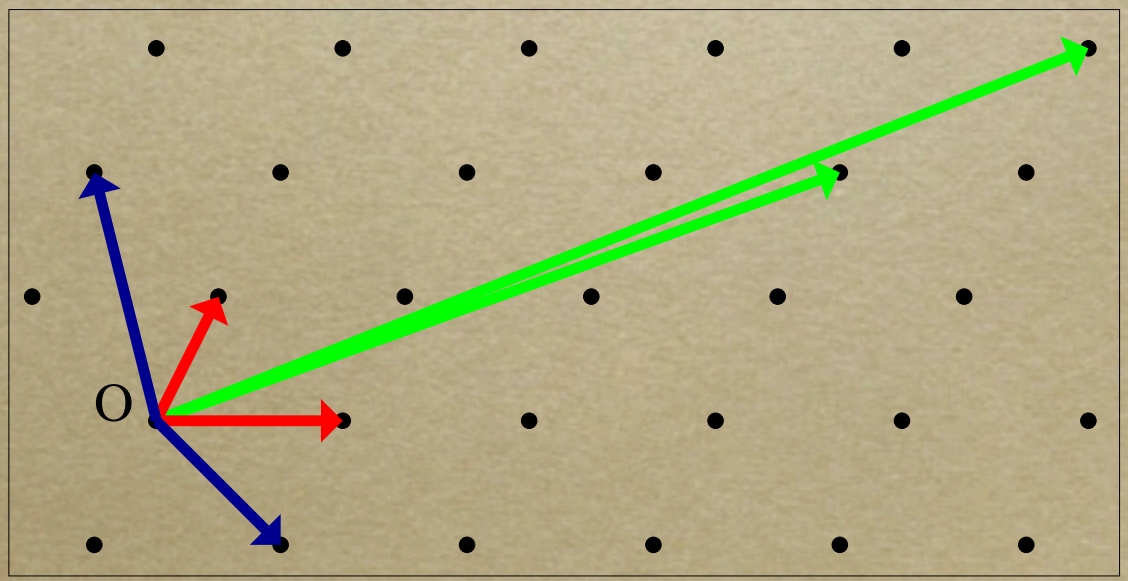


- A lattice is infinite, but lattice crypto actually uses the finite abelian group $\mathbf{Z}^d/L$: it works modulo the lattice L.

Lattice Invariants

- The dim is the dim of span(L).
- The (co-)volume is the volume of any basis parallelepiped: can be computed in poly-time.

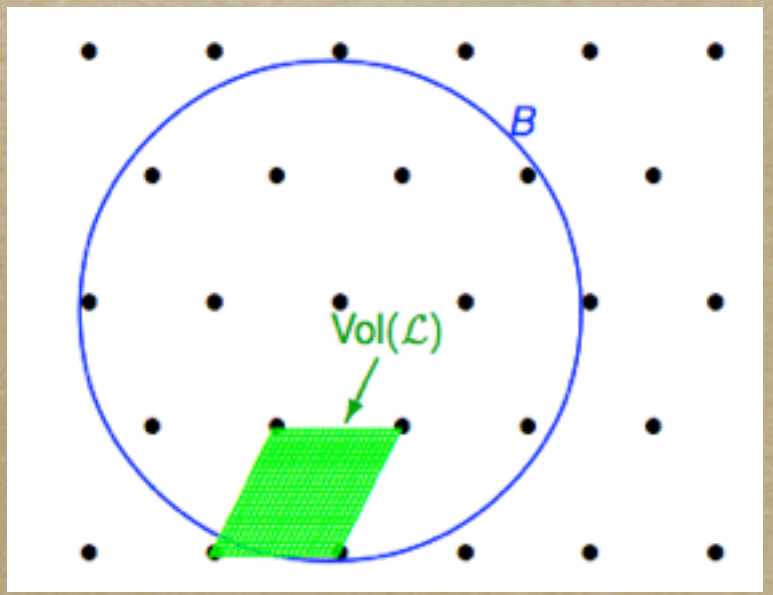
  Ex: vol($\mathbf{Z}^n$)=1.



The Gaussian Heuristic

- The volume measures the density of lattice points.
- For "nice" full-rank lattices L, and "nice" measurable sets C of $\mathbf{R}^n$:

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

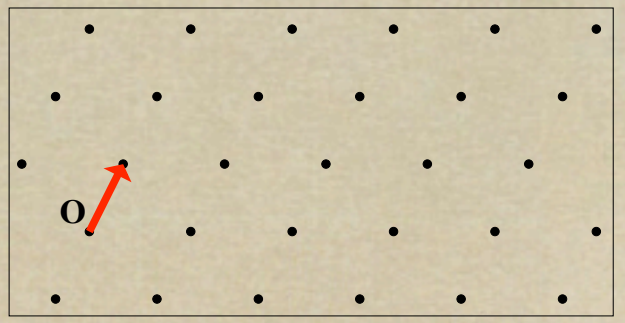


Volume of Balls

- The volume of the n-dim ball of radius R is:

$$V_n(R) = \frac{\pi^{n/2}}{\Gamma(1+n/2)} R^n \sim \frac{1}{\sqrt{n\pi}} \left(\frac{2\pi e}{n}\right)^{n/2} R^n$$

Short Lattice Vectors

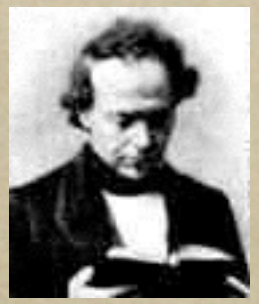


- Any d-dim lattice L has exponentially many vectors of norm ≤

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

- In a random d-dim lattice L, all non-zero vectors have norm ≥

$$\Omega\left(\sqrt{d}\right)\mathrm{vol}(L)^{1/d}$$



Hermite's Constant (1850)

- This is the "worst-case" for short lattice vectors.
- Hermite showed the existence of this constant:

$$\sqrt{\gamma_d} = \max_L \frac{\min_{\vec{v} \in L, \vec{v} \neq 0} \|\vec{v}\|}{\mathrm{vol}(L)^{1/d}}$$

Lattices and Complexity

- Since 1996, lattices are very trendy in complexity (STOC/FOCS): classical and quantum.
- Depending on the approximation factor with respect to the dimension:
  - NP-hardness
  - non NP-hardness (NP∩co-NP)
  - worst-case/average-case reduction
  - cryptography
  - polynomial-time algorithms

1

$O(1)$

$\sqrt{n}$

$O(n \log n)$

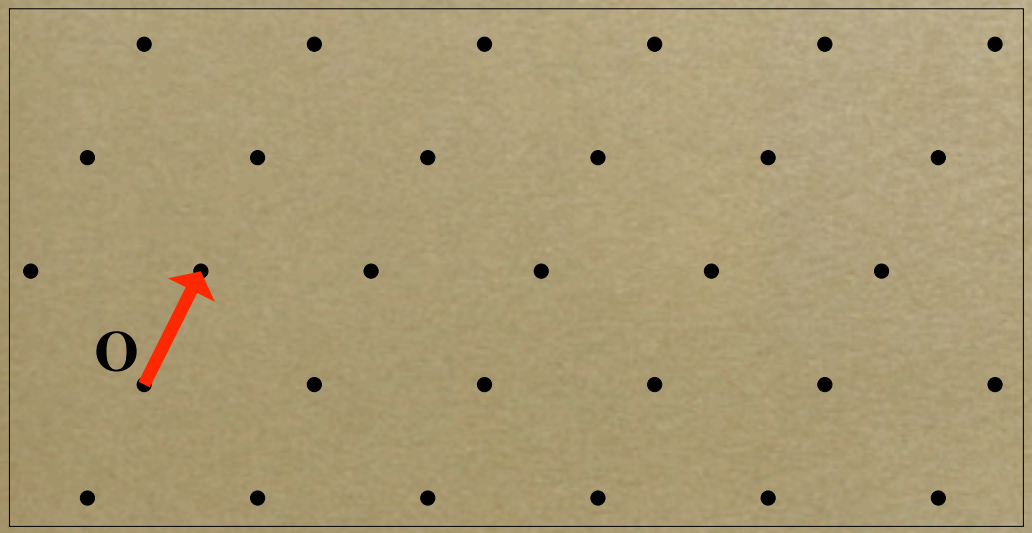
$n^{O(1)}$

$2^{O(n \log \log n / \log n)}$

∞

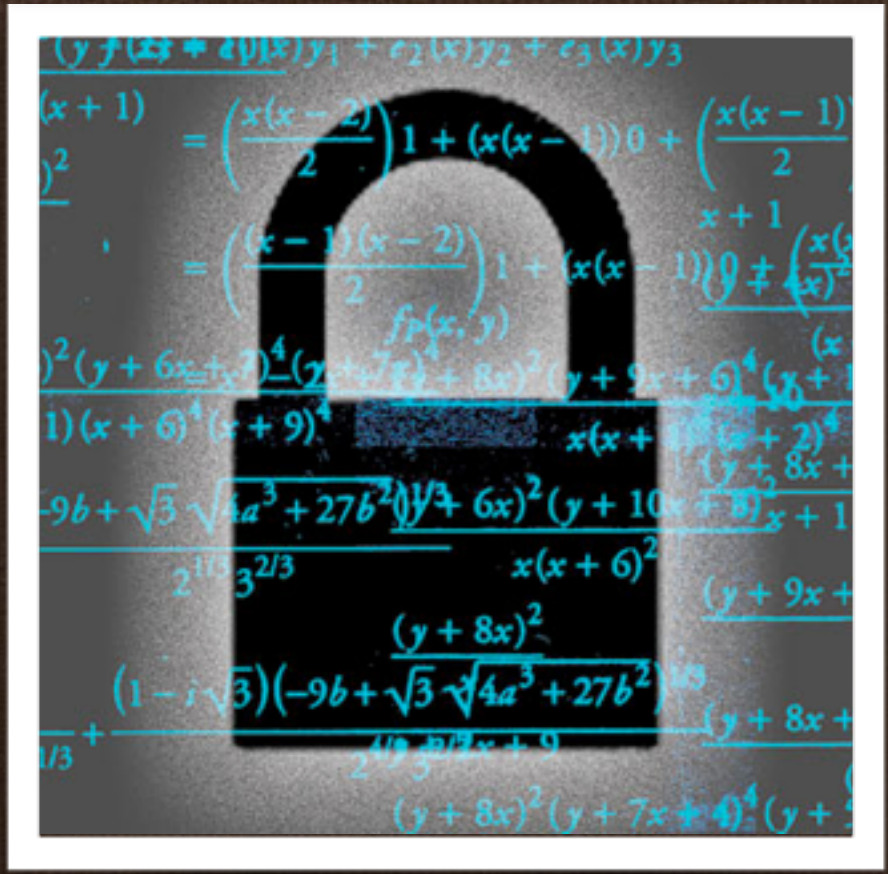
The Shortest Vector Problem (SVP)

- Input: a basis of a d-dim lattice L
- Output: nonzero $v \in L$ minimizing $\|v\|$.
- Approx: $\|v\| \le f(d)\ \|w\|$ for all non-zero $w \in L$.



2	0	0	0	0
0	2	0	0	0
0	0	2	0	0
0	0	0	2	0
1	1	1	1	1

Lattice-based Cryptography



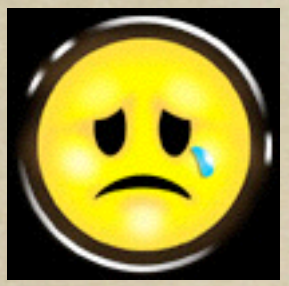
Lattice-based Cryptography

- **RSA** uses large finite (abelian) groups $G=(\mathbb{Z}/N\mathbb{Z})^\times$ (2048 bits, 4096 bits...). To speed things up:
  - **Elliptic curve crypto** uses smaller groups, whose operations are more expensive.
  - **Lattice cryptography** uses larger groups, but whose operations are much cheaper.

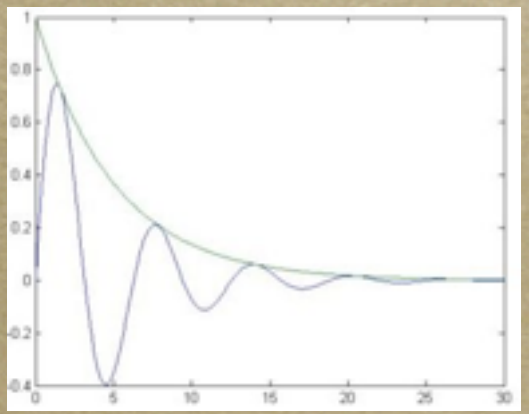
Lattice-based Cryptography: the Pros

- Can be more efficient than RSA SECURITY
- Potentially resistant to quantum computers. Several groups are working on a lattice-TLS.
- Can have security properties based on worst-case assumptions.
- Very trendy. Recent breakthroughs: fully homomorphic encryption, multilinear maps and obfuscation.

Lattice-based Cryptography: the Cons

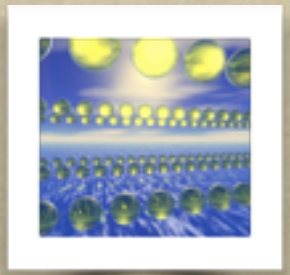


- Apart from NTRU, few concrete proposals of parameter sets: practicality is often unclear.
- Gain might only be asymptotic.



Lattices in Cryptology

- Three years stand out:
  - 1982: First use of lattices in cryptanalysis: knapsack cryptosystems.
  - 1996: First crypto schemes based on hard lattice problems: NTRU, Ajtai-Dwork, GGH...
  - 2009: Fully-homomorphic encryption based on lattices.



Lattice-based Crypto

- Somewhat a revival of knapsack crypto (MerkleHellman78...)
- Two Families:
  - "Theoretical": [Ajtai96...] focus on security proofs.
  - "Applied": [NTRU96...] focus on efficiency.
- They "interact" more and more: [Micc02,GPV08,Gentry09,Peikert10,MiPe12...]

Lattice Problems in Crypto

- In many crypto schemes, one actually deals with problems not defined using lattices:
  - SIS.
  - LWE.
- Both are connected to lattice problems and usually presented with linear algebra: instead, we adopt a group-theoretical point of view, to clarify the use of duality.

The SIS Problem (Ajtai1996) [Small Integer Solution]

- Let $(G,+)$ be a finite Abelian group:

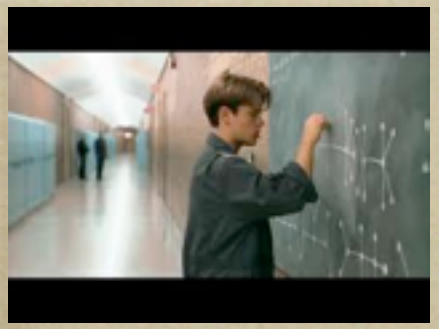
  [Ajtai96] used $G=(\mathbf{Z}/q\mathbf{Z})^n$.

- Pick $g_1,\ldots,g_m$ uniformly at random from G.

- Goal:

  Find short $x=(x_1,\ldots,x_m)\in\mathbf{Z}^m$ s.t. $\sum_i x_i g_i = 0$.

- This is essentially finding a short vector in a (uniform) random lattice of $L(G)$ = { lattices $L\subseteq\mathbf{Z}^m$ s.t. $\mathbf{Z}^m/L \sim G$ }



Worst-case to Average-case Reduction

- [Ajtai96]: If one can efficiently solve SIS for $G=(\mathbf{Z}/q\mathbf{Z})^n$ on the average, then one can efficiently find short vectors in every n-dim lattice.
- [GINX16]: This can be generalized to any finite abelian group G, provided that #G is sufficiently large $\geq n^{\Omega(\max(n,\mathrm{rank}(G)))}$

  Note: $(\mathbf{Z}/2\mathbf{Z})^n$ is not.

Duality

- A character of G is a morphism from G to the torus $\mathbf{T}=\mathbf{R}/\mathbf{Z}$.
- G is isomorphic to its dual group G^x = {characters of G}.
- The dual lattice of the SIS lattice

  $L=\{x=(x_1,\dots,x_m)\in\mathbf{Z}^m$ s.t. $\sum_i x_i g_i = 0\}$ is

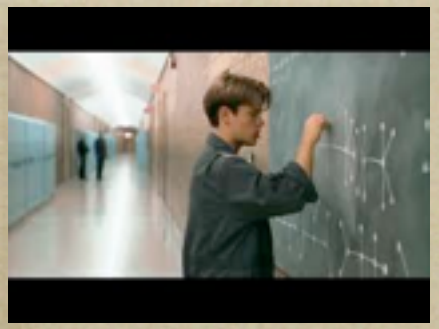
  $L^x=\{y=(y_1,\dots,y_m)\in\mathbf{R}^m$ s.t. $y_i\equiv s(g_i) \pmod 1$ for some $s\in G^x\}$

The LWE Problem (Regev2005) [Learning with Errors]

- Let (G,+) be a finite Abelian group:
  [Regev05] used $G=(\mathbf{Z}/q\mathbf{Z})^n$ like [Ajtai96].
- Pick $g_1,\ldots,g_m$ uniformly at random from G.
- Pick a random character s in G^x.
- Goal: recover s given $g_1,\ldots,g_m$ and noisy approximations of $s(g_1),\ldots, s(g_m)$, where the noise is Gaussian.

Ex: Cyclic G

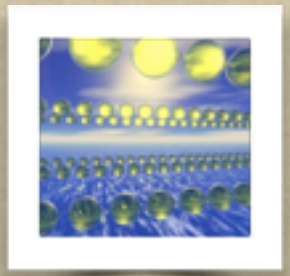
- Let G = $\mathbf{Z}/q\mathbf{Z}$
- Pick $g_1,\ldots,g_m$ uniformly at random mod q.
- Goal: recover s given $g_1,\ldots,g_m$ and randomized approximations of sg_1 mod q,..., sg_m mod q.
- This is exactly a randomized variant of Boneh-Venkatesan's Hidden Number Problem from CRYPTO '96.



Worst-case to Average-case Reduction

- [Regev05]: If one can efficiently solve LWE for $G=(\mathbf{Z}/q\mathbf{Z})^n$ on the average, then one can quantum-efficiently find short vectors in every n-dim lattice.
- [GINX16]: This can be generalized to any finite abelian group G, provided that #G is sufficiently large.

Lattice Cryptography: Design



Lattice-based Crypto

- Two Types of Techniques
  - Cryptography using trapdoors, i.e. secret short basis of a lattice. Similarities with RSA/Rabin cryptography.
  - Cryptography without trapdoors. Similarities with DL cryptography.
- Case study: Encryption.

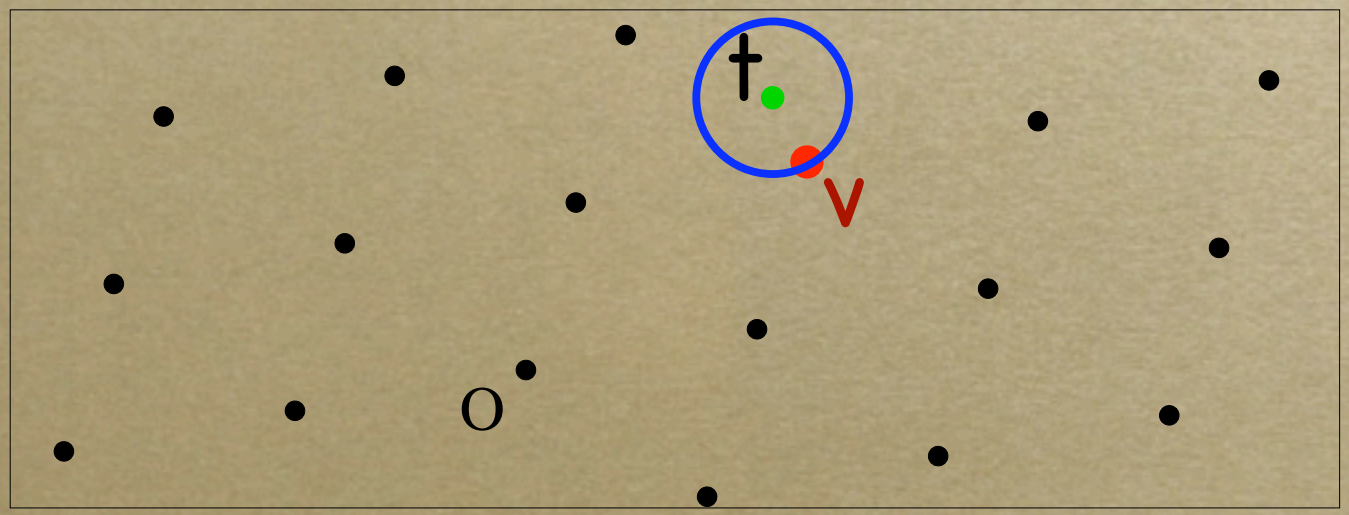
Trapdoor-based Encryption

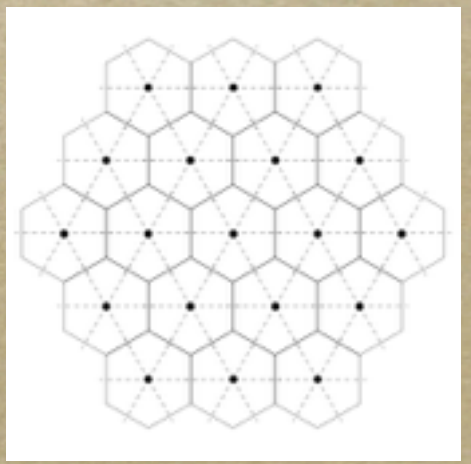
Remember

- N=pq product of two large random primes.
- $ed \equiv 1 \pmod{\phi(N)}$ where $\phi(N)=(p-1)(q-1)$.
  - e is the public exponent
  - d is the secret exponent
- Then $m \to m^e$ is a trapdoor one-way permutation over $\mathbf{Z}/N\mathbf{Z}$, whose inverse is $c \to c^d$.

Bounded Distance Decoding (BDD)

- Input: a basis of a lattice L of dim d, and a target vector t very close to L.
- Output: $v \in L$ minimizing $\|v-t\|$. Easy if one knows a nearly-orthogonal basis.





Reducing Modulo a Lattice

- If L is an integer lattice, the quotient $\mathbf{Z}^n/L$ is a finite group, with many representations: lattice crypto works modulo a lattice.
- We call L-reduction any efficiently computable map f from $\mathbf{Z}^n$ s.t. f(x)=f(y) iff $x-y \in L$.

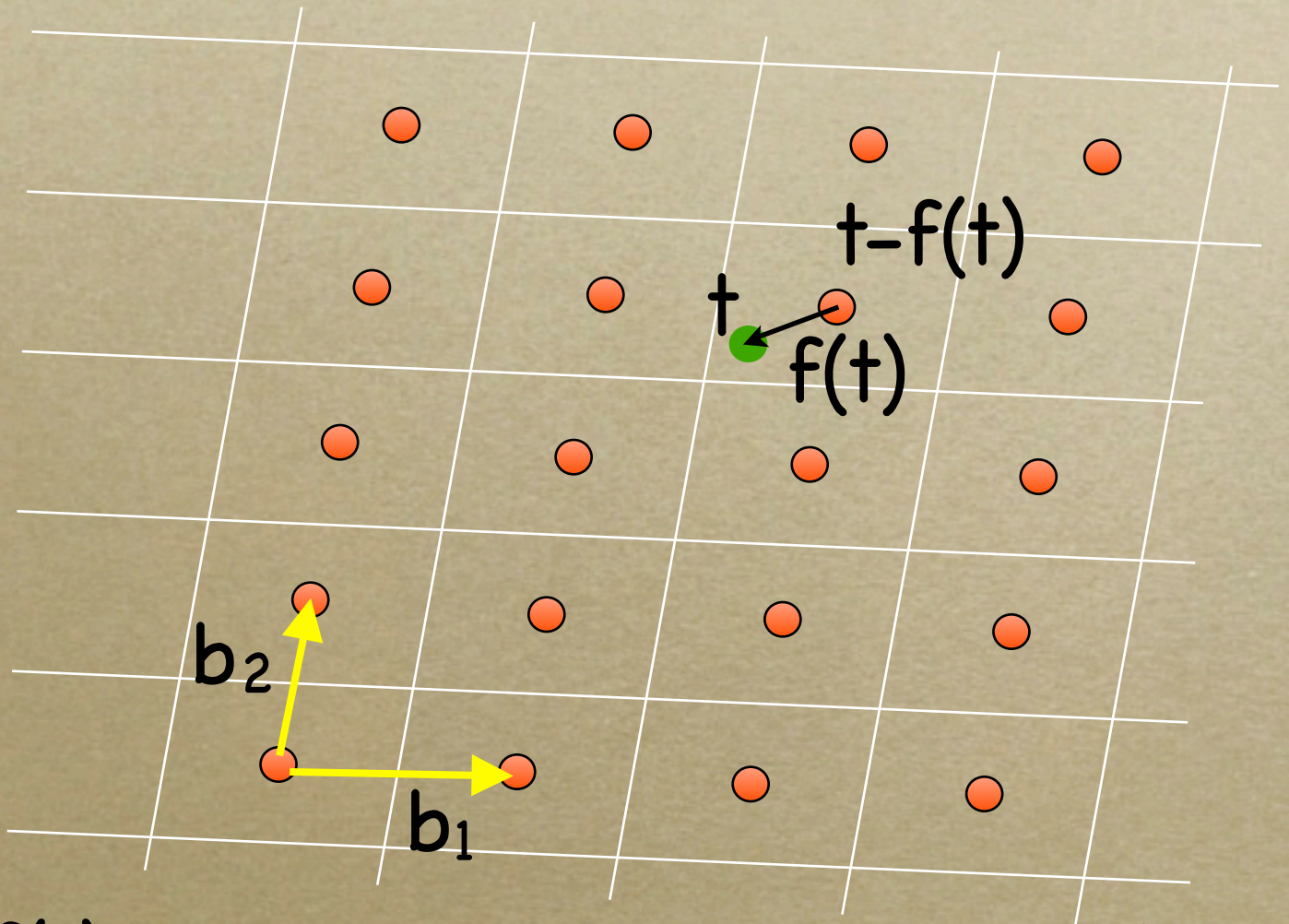
One-Way Functions from BDD

- If BDD is hard, any public L-reduction f is a one-way function.
  - Let (t,L) be a BDD instance: t=v+e where $v \in L$ and e is very short.
  - Then f(t)=f(e) because $t-e \in L$: if f is not one-way, then given f(e), one can recover e and also the BDD solution v=t-e.

Building L-Reductions

- Any basis provides two L-reductions, thanks to Babai's nearest plane algorithm and rounding-off algorithm.
- NTRU encryption implicitly uses a L-reduction.

Ex: Babai's rounding off



Choose f(t) in the basis parallelepiped s.t. t-f(t)∈L

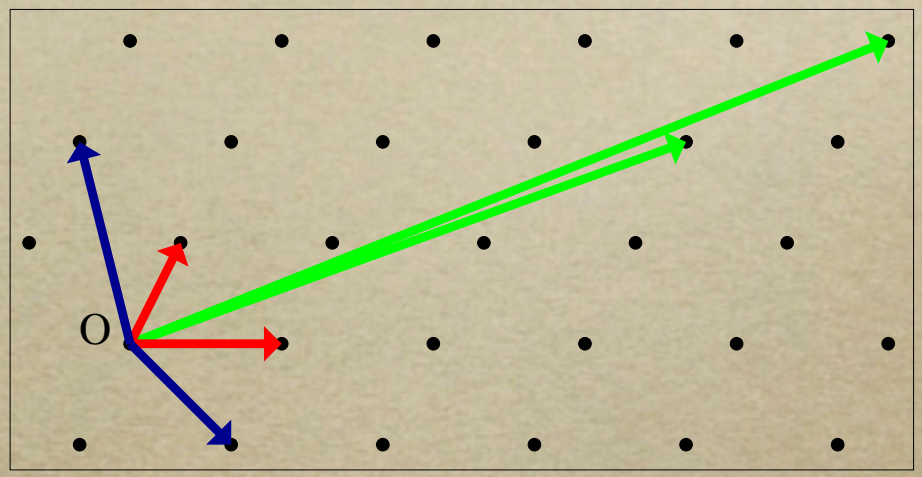
Ex: NTRU Encryption

- Ring $R=\mathbf{Z}[X]/(X^N-1)$, secret key=short $(f,g)\in R^2$, public key $h=g/f \pmod q$.
- A message m is a short element of R.
- Its ciphertext is $c=m+pr*h \pmod q$ where r is a sparse element of R. This corresponds to the L-reduction $F(m,-r) = (m+pr*h \pmod q, 0)$ for the lattice $L=\{(u,v)\in R^2 \text{ s.t. } u\equiv pv*h \pmod q\}$

Solving BDD by L-reduction

- The L-reductions derived from Babai's algorithms leave some set invariant: there exists $D(B) \subseteq \mathbf{Z}^n$ s.t. $f(x)=x$ for all $x \in D(B)$. This allows to solve BDD when the error$\in D(B)$.
- The largest ball inside D(B) depends on the quality of the basis.

Deterministic Public-Key Encryption [GGH97-Micc01]



- Secret key = Good basis
- Public key = Bad basis
- Message = Short vector
- Encryption = L-reduction with the public key
- Decryption = L-reduction with the secret key

Optimization: NTRU Encryption

- Ring $R=\mathbf{Z}[X]/(X^N-1)$, secret key $(f,g)\in R^2$, public key $h=g/f \pmod q$.
- Encryption can be viewed [Mi01] as L-reducing a short vector with the Hermite normal form, where $L=\{(u,v)\in R^2$ s.t. $u\equiv pv*h \pmod q\}$.
- Decryption is a special BDD algorithm using the secret key (f,g).

Trapdoor-less Encryption

Diffie-Hellman Key Exchange

- Let $G = \langle g \rangle$ be generated by g of order q.

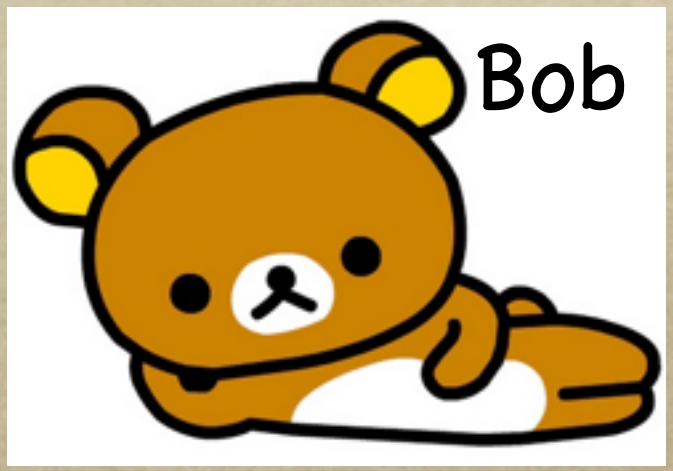
Alice

$a \in_R \mathbf{Z}/q\mathbf{Z}$

$g^a \in G$ →

← $g^b \in G$

$b \in_R \mathbf{Z}/q\mathbf{Z}$



- Both can compute the shared key g^{ab}.
- This key exchange is the core of El Gamal public-key encryption.

Abstracting DH

- Let e: $(a,b) \mapsto g^{ab}$. This map is a pairing: it is bilinear.

$$\mathbf{Z}_q \times \mathbf{Z}_q \rightarrow G$$

- Let f: $a \mapsto g^a$ be the DL one-way function

$$\mathbf{Z}_q \rightarrow G$$

- e(a,b) can be computed using (f(a),b) or (a,f(b)), i.e. even if a or b is hidden by f.
- Security = hard to distinguish (f(a),f(b),e(a,b)) from (f(a),f(b),random). This is called DDH.

DH with Lattices?

- What would be the pairing?
- What would be the one-way function to hide inputs?

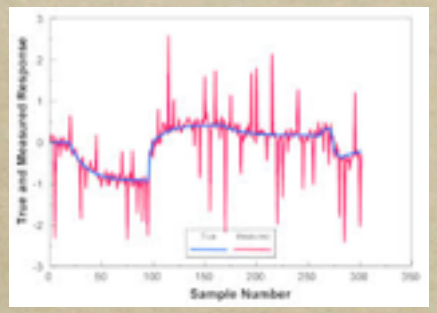
Pairing from Lattices

- Let $g_1,\ldots,g_m$ in G. The dual group G^x induces a pairing $G^x \times \mathbf{Z}^m \to \mathbf{R/Z}$ by $\varepsilon(s,(x_1,\ldots,x_m)) = s(\sum_i x_i g_i)$

- Let $y=f_g(x_1,\ldots,x_m)=\sum_i x_i g_i \in G$ where x_i's small.

  and $b=f^x_g(s,e)= (s(g_1),\ldots,s(g_m))+e \in (\mathbf{R/Z})^m$, e small.

- Then $\varepsilon(s,(x_1,\ldots,x_m))$ can be computed from (s,y) or $(b,(x_1,\ldots,x_m))$ as $s(\sum_i x_i g_i) = \sum_i x_i s(g_i) \approx \langle (x_1,\ldots,x_m),b \rangle$ because the x_i's are small.



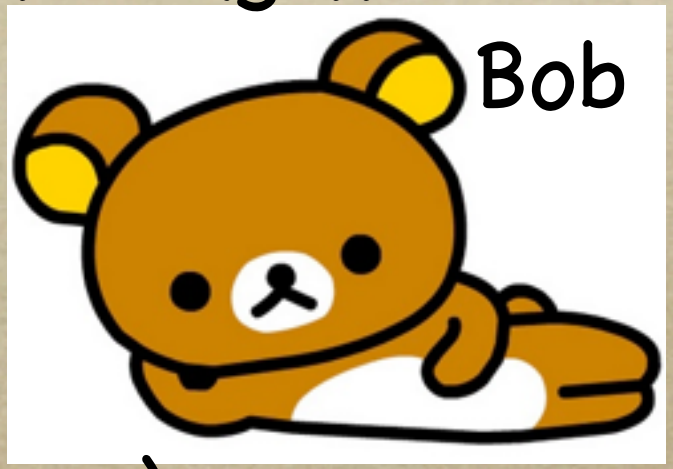
Noisy Key-Exchange from Lattices

- Let $g_1,\ldots,g_m$ generate G.

$$b=f^x_g(s,e)=(s(g_1),\ldots,s(g_m))+e$$

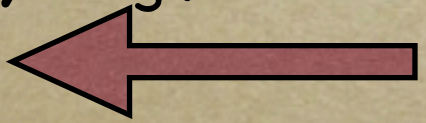
Alice

$s\in_R G^x$



Bob

$y=f_g(x_1,\ldots,x_m)=\sum_i x_i g_i$

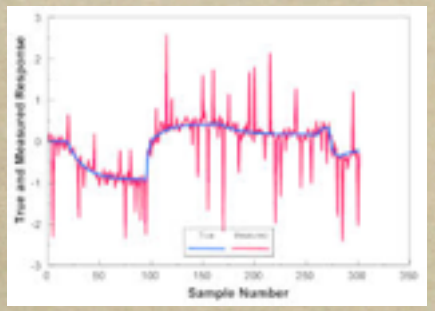


short $(x_1,\ldots,x_m)$

- Both compute an approx of $\varepsilon(s,(x_1,\ldots,x_m))=s(y)$:

Alice computes $s(y)+e'$ and

Bob computes $\sum_i x_i b_i$.



El Gamal Encryption from Lattices

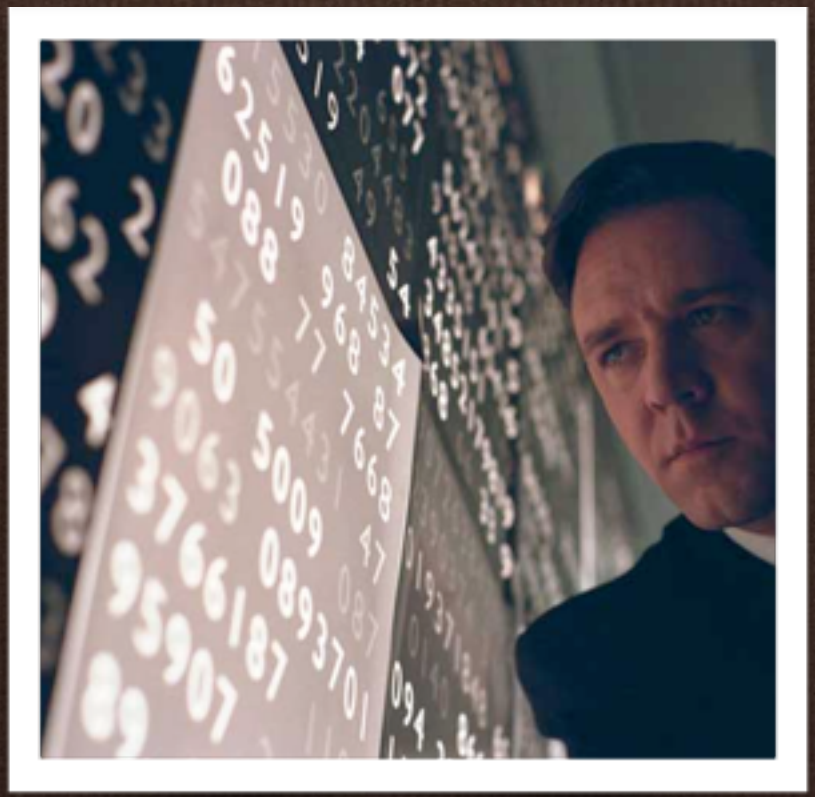
- This key exchange gives rise to two El Gamal-like public-key encryption schemes, because the lattice pairing is not symmetric.
- These El-Gamal-like schemes are IND-CPA-secure under the hardness of LWE/SIS.
- Similarly, many LWE/SIS schemes can be viewed as analogues of the RSA/DL world: [GPV08] is a lattice analogue of Rabin's signature, etc.

Concrete Lattice Schemes

- Encryption
  - NTRU [HPS1998...]: Parameters have changed several times (Ex: decryption failure attacks), but the core idea remains the same. Has been standardized.
  - RLWE schemes: more efficient than LWE, but stronger hardness assumption. Being used in lattice-TLS prototypes.

Concrete Lattice Schemes

- Signature
  - NTRU: Less succesful than encryption. The latest version is NTRU-MLS [PQC '14], after deadly attacks on NSS [GJSS2001,GeSz02] and NTRUSign [NgRe2006,DuNg12].
  - BLISS [DDLL2013]: the most optimized version uses NTRU-like assumptions.
  - For now, Fiat-Shamir signatures are more efficient than Hash-and-Sign signatures...



Cryptanalysis of Lattice-based Cryptography

SVP Algorithms

- Polynomial-time approximation algorithms.
  - The LLL algorithm [1982].
  - Block generalizations by [Schnorr1987], [GHKN2006], [Gama-N2008], [MiWa2016].
- Exponential exact algorithms.
  - Polynomial-space: Enumeration [Kannan1983] and pruning variants etc.
  - Exponential-space: Sieving [AKS2001], [MiVo10]

Maths vs Algorithms

- Maths: Proving the existence of short lattice vectors i.e. upper bounds on Hermite's constant.
- Algorithms: Finding short or shortest lattice vectors.

Maths vs Algorithms

- Three mathematical inequalities have been turned into efficient algorithms.
  - Hermite's inequality: the LLL algorithm.
  - Mordell's inequality: blockwise algorithms.
  - Minkowski's inequality (Mordell's proof)
    - Worst-case to average-case reductions
    - Sieve algorithms [ADRS15]

1773
Lagrange

Hermite
1850

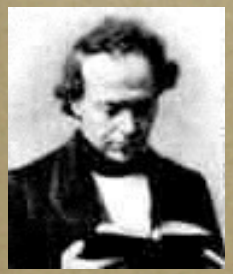
1982

Hermite's Inequality and the LLL Algorithm

Short Lattice Vectors

- [Lagrange1773]: In any 2-dim lattice L, there is a nonzero vector of norm $\leq (4/3)^{1/4}\ \mathrm{vol}(L)^{1/2}$.

$$\sqrt{\gamma_2} = \left(\frac{4}{3}\right)^{1/4}$$



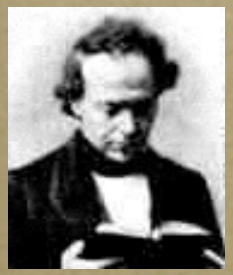
- [Hermite1850]: In any d-dim lattice, there is a nonzero vector of norm $\leq (4/3)^{(d-1)/4}\ \mathrm{vol}(L)^{1/d}$

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

Finding Short Lattice Vectors

- [Lagrange1773]'s algorithm efficiently outputs a nonzero vector of norm $\leq (4/3)^{1/4}\ \mathrm{vol}(L)^{1/2}$.

$$\sqrt{\gamma_2} = \left(\frac{4}{3}\right)^{1/4}$$



- [Hermite1850] gives an implicit (inefficient) algorithm to output a vector of norm $\leq (4/3)^{(d-1)/4}\ \mathrm{vol}(L)^{1/d}$

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

The Lenstra-Lenstra-Lovász Algorithm (1982)

- Th: Given $\varepsilon > 0$ and a d-dim lattice L, [LLL82] finds, in time polynomial in size(lattice) and $1/\varepsilon$, a basis whose 1st vector satisfies:

$$\|b_1\| \leq (4/3+\varepsilon)^{(d-1)/4}\ \mathrm{vol}(L)^{1/d}$$
$$\|b_1\| \leq (4/3+\varepsilon)^{(d-1)/2}\ \|L\|$$

- LLL is an algorithmic version of Hermite's inequality:

$$\|L\| \leq \left(\frac{4}{3}\right)^{(d-1)/4} \mathrm{vol}(L)^{1/d}$$

Intuition

- Hermite's inequality is based on two ideas:
  - Projection: this creates a lower-dimensional lattice.
  - Lifting short projected vectors
- LLL adds relaxation to guarantee polynomial time.

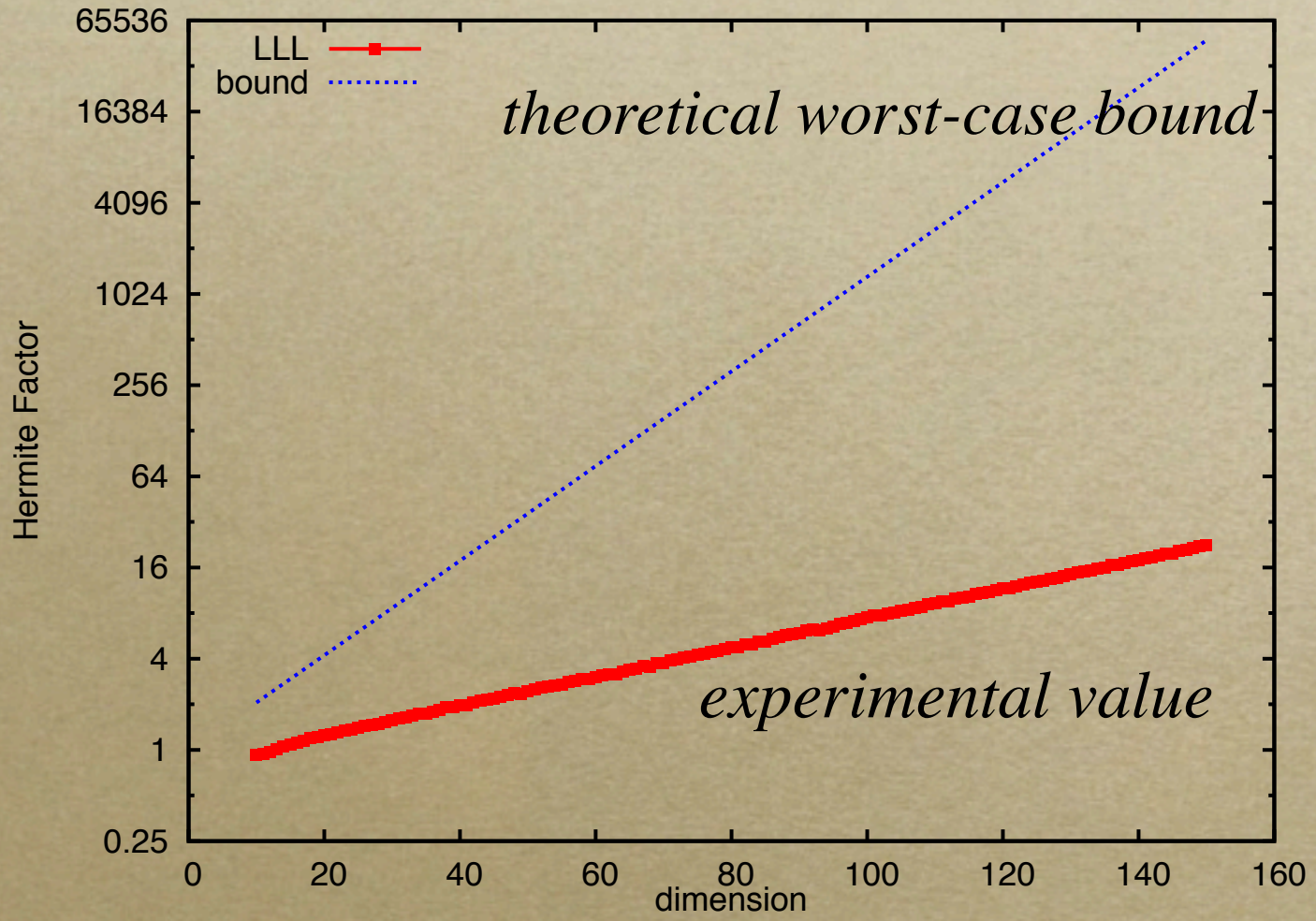
The Magic of LLL

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

LLL: Theory vs Practice

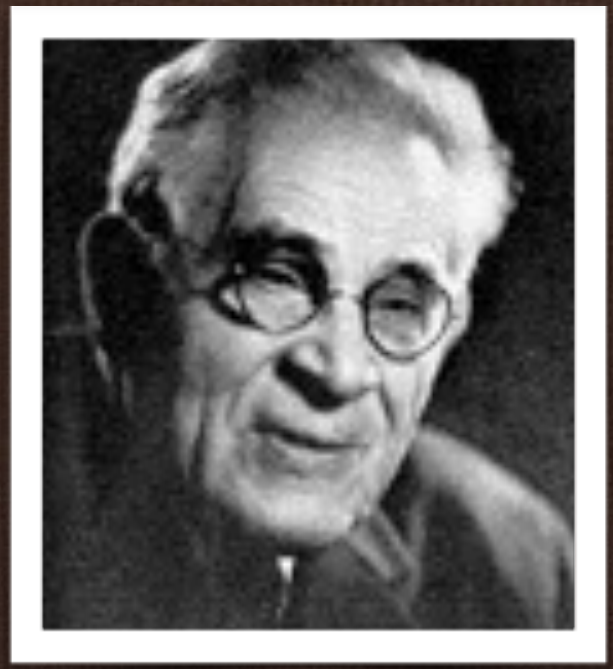
- The approx factors $(4/3+\varepsilon)^{(d-1)/4}$ and $(4/3+\varepsilon)^{(d-1)/2}$ are tight in the worst case: but this is only for worst-case bases of certain lattices.
- Experimentally, $4/3+\varepsilon \approx 1.33$ can be replaced by a smaller constant ≈ 1.08, for any lattice, by randomizing the input basis.
- But there is no good explanation for this phenomenon, and no known formula for the experimental constant ≈ 1.08.

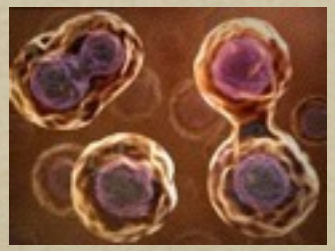
Illustration



Log(Hermite Factor)

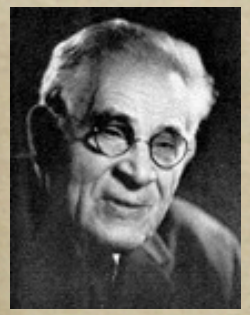
Mordell's Inequality and Blockwise Algorithms





Divide and Conquer

- LLL is based on a local reduction in dim 2.
- Blockwise algorithms find shorter vectors than LLL by using an exact SVP-subroutine in low dim k called the blocksize.
  - Even if the subroutine takes exponential time in k, this is polynomial in d if k=log d.



Mathematical Analogy

- If we show the existence of very short lattice vectors in dim k, can we use it to prove the existence of very short lattice vectors in dim d > k?
- [Mordell1944]'s inequality generalizes Hermite's inequality:

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

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

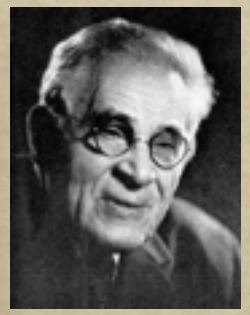
Approximation Algorithms

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

$$\|L\| \leq \left(\frac{4}{3}\right)^{(d-1)/4} \mathrm{vol}(L)^{1/d} = \sqrt{\gamma_2}^{d-1} \mathrm{vol}(L)^{1/d}$$

- The [GamaN08] algorithm is an algorithmic version of [Mordell1944]'s inequality.

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



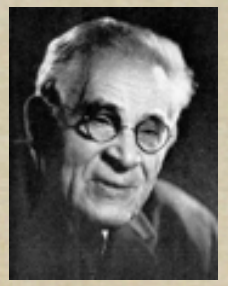
Mordell's Inequality (1944)

- Hermite's inequality is the k=2 particular case of Mordell's inequality:

$$\gamma_d \leq \gamma_k^{(d-1)/(k-1)} \text{ if } 2 \leq k \leq d$$

- All known proofs of Mordell's inequality are based on duality: the Gama-N algorithm also uses duality, which provides a different way to decrease the lattice dimension.

An Algorithmic Version of Mordell's Inequality



- The algorithm of [GamaN2008] solves Hermite-SVP with factor essentially $\sqrt{\gamma_k}^{(d-1)/(k-1)}$ using a k-dim SVP-oracle.
- This algorithm is to Mordell's inequality what LLL is to Hermite's inequality.
- By choosing an appropriate k=log d, the whole algorithm is poly-time with a subexponential approx factor.

Security Estimates

In the 1980s-1990s

- Lattice algorithms were somewhat a dark art.

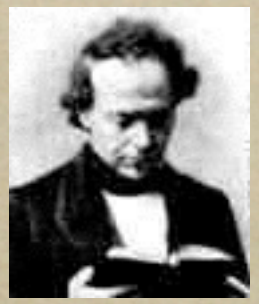
- It was noticed that algorithms performed better than theoretically expected, but it was unclear by how much exactly.

Issues

- Theory is usually insufficient: worst-case analyses do not match experiments.
- For crypto, we want more than a worst-case bound, we would like to ``predict'' the running times.
  - Much more difficult
  - Analogy with physics

Security Estimates

- Somewhat independent of security proofs
- Identify the best attack based on the state-of-the art
  - Find as many attacks as possible
  - Identify the ``best'' one
  - Select keysizes/parameters accordingly



A Core Problem

- To assess the cost of a lattice attack, it is useful to reduce it to a core problem.
- The most popular core problem is the Hermite-Approx-SVP problem:
  - Given a basis of a n-dim lattice L and an approximation factor f(n), find a non-zero v of norm $\leq f(n)\ \mathrm{vol}(L)^{1/n}$.

Solving Lattice Problems In Practice

- The Hermite-factor is convenient:
  - Run the algorithm on a random lattice.
  - Measure $\frac{\|\vec{b}_1\|}{\mathrm{vol}(L)^{1/d}}$, typically exp. in d.
- Performances for the main lattice problems (SIS, LWE, etc.) can be derived [GaN08, MiRe09...]. But maybe not so clear for NTRU.

Estimating Lattice Attacks

- Given some lattice attack, it is often possible to (roughly) estimate its efficiency
  - Assess the required Hermite-factor
    - If LLL is enough, assess cost(LLL)
    - Otherwise, assess the required BKZ-blocksize and its cost
  - We only require an order of magnitude.

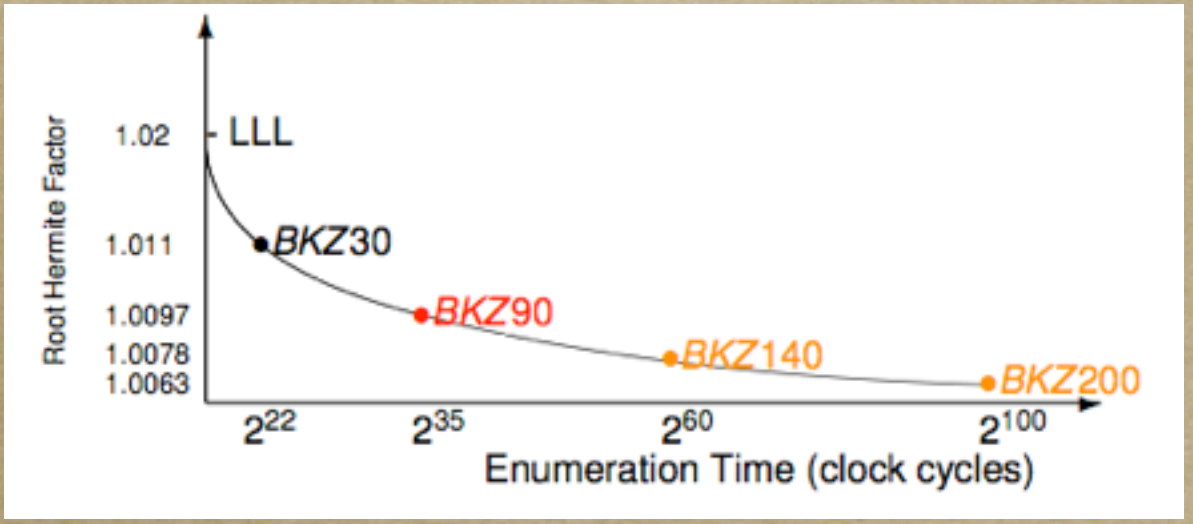
Predicting BKZ [ChN11]

- Predict the behaviour of high-blocksize state-of-the-art BKZ ($k \geq 50$), using an efficient simulation algorithm: the minimum of most k-dim blocks seems to behave like random lattices.

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

becomes roughly

$$\mathrm{GaussHeurist(k)}^{(d-1)/(k-1)}$$



Analogy with Factoring

- When analyzing sieve algorithms (QS, NFS, etc.), we assume heuristically that certain numbers behave like random numbers, when it comes to smoothness probability.
- Here, we assume that certain (projected) lattices behave like random lattices, when it comes to the first minimum.

Illustration for Average Dim

Approx Blocksize	Target H-factor
85	1.01^d
106	1.009^d
133	1.008^d
168	1.007^d
216	1.006^d
286	1.005^d

Are We Done?

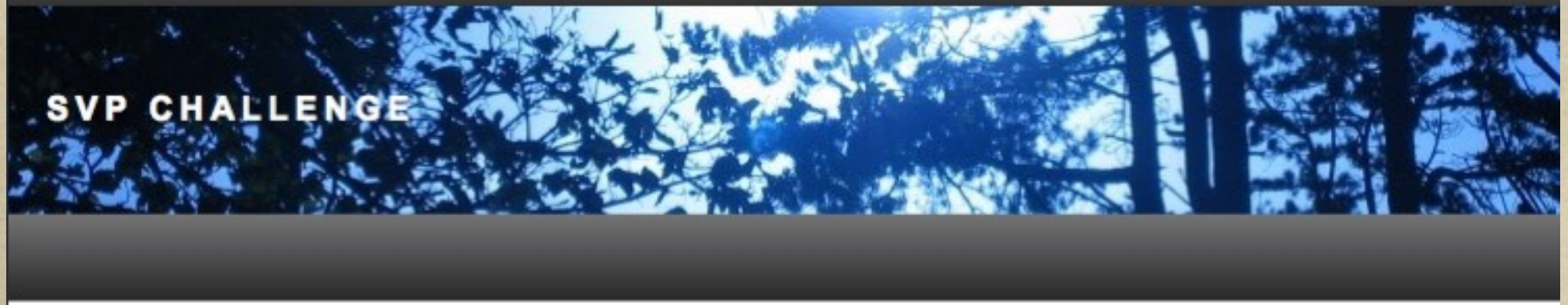
- It remains to estimate the cost of the BKZ-subroutine: finding (nearly-)shortest vectors in ``small'' dimensions...
- This is where things are not so clear: optimization is difficult and there has been progress in the past few years.

In a Nutshell

- In current implementations, the SVP-subroutine for BKZ is extreme-pruning enumeration [GNR10].
- There are "reasonable" predictions [GNR10] for this subroutine, but optimization is unclear: lower and upper bounds in [ChN11].

Approx Blocksize	Cost in core-days
140	$\leq 2^{23}$
170	$\leq 2^{49}$
250	$\leq 2^{135}$

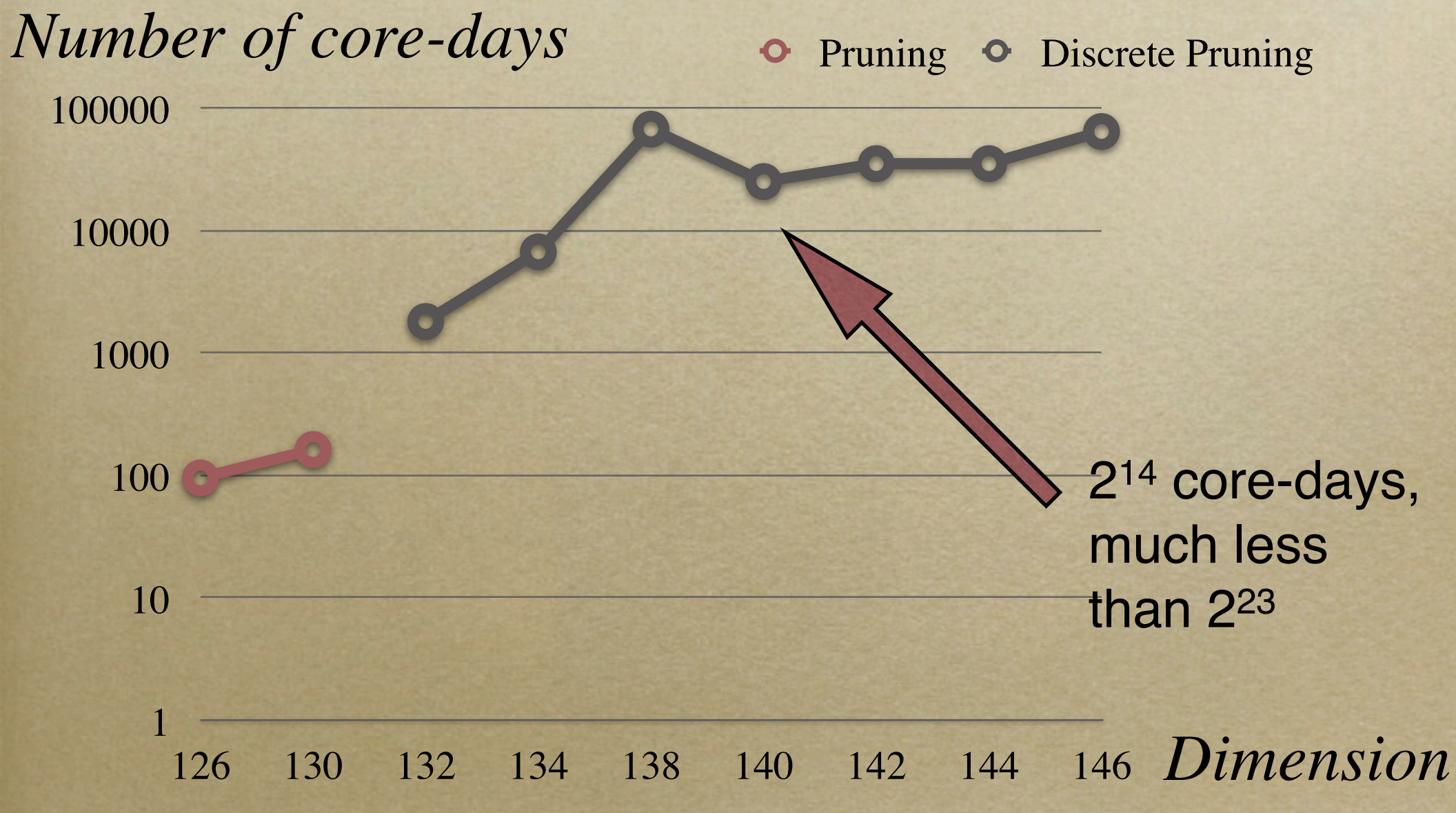
[ChN11] Figures



HALL OF FAME

Position	Dimension	Euclidean Norm	Seed	Contestant	Solution	Algorithm	Subm. Date	Approx. Factor
1	146	3195	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2015-08-24	1.04534
2	144	3154	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2015-06-21	1.04284
3	142	3141	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2015-03-15	1.04609
4	140	3025	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2015-01-23	1.01139
5	138	3077	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2014-12-7	1.03516
6	134	2976	0	Kenji KASHIWABARA and Tadanori TERUYA	vec	Other	2014-07-13	1.01695
7	132	3012	0	Kenji Kashiwabara and Masaharu Fukase	vec	Other	2014-04-24	1.03787
8	130	2883	0	Yoshinori Aono and Phong Nguyen	vec	ENUM,BKZ	2014-10-9	0.99871
9	130	3025	0	Kenji Kashiwabara and Masaharu Fukase	vec	Other	2013-11-15	1.04787
10	128	2984	0	Kenji Kashiwabara and Masaharu fukase	vec	Other	2013-09-23	1.04017
11	128	2992	0	Kenji Kashiwabara and Masaharu Fukase	vec	Other	2013-09-19	1.04313
12	126	2855	0	Yoshinori Aono and Phong Nguyen	vec	ENUM,BKZ	2014-09-9	1.00556

The SVP Challenges



Comparison with RSA

- The largest computation for the SVP challenges is for dim 138 :
  66,000 core-days ≈ 2^{63} clock cycles.
- This is 11 times cheaper than RSA-768 =
  730,000 core-days ≈ 2^{67} clock cycles.
- The 140-dim computation confirms that the upper bounds of [ChN11] are only upper bounds.

State-of-the-Art

- We understand reasonably well the best lattice algorithms: we can guess the approximate quality and the approximate running time for a given set of parameters.
- But we don't know well how to optimize the selection of parameters, and there might be improvements for the subroutine: the latest SVP records use a different form of pruned enumeration.

Impact

- We should be conservative: security margin. Fortunately, only a major improvement in blocksize can impact H-factor estimates: an order of magnitude for the blocksize is sufficient.
- Security estimates are tricky, especially for high security levels like 256-bit security: already non-trivial for RSA.

The Importance of Numerical Challenges

- Challenges are very useful to check the state-of-the-art

NTRU Challenges

Conclusion

Design

- RSA/DL works in certain finite (abelian) groups G.
- Lattice cryptography works in (abelian) finite quotient groups $G' = \mathbf{Z}^n/L$.
  - The trapdoor is a secret representation of G'.
  - G' may be much bigger than G, but has cheaper operations.
  - Noise and small preimages require to adapt RSA/DL constructions.

Design

- Lattice cryptography is different from classical public-key cryptography, but there are also many similarities.
  - Making analogies hopefully helps our understanding.

Trends in Design

- More and more classical PK schemes have been adapted to the lattice setting.
  - Can all be adapted?
- A few lattice schemes achieve new functionalities (FHE, multilinear maps, general attribute-based encryption).
  - Can they be achieved without lattices?

Cryptanalysis

- There has been significant progress in lattice algorithms in the past 10 years.
  - It is a positive sign that the problem is attracting more and more attention.
  - On the other hand, how are we going to model future progress in security estimates?

Quantum Cryptanalysis

- There are very few examples of quantum algorithms... especially in cryptanalysis.
- Until we have a quantum computer to play with, it will be difficult to know the true power of quantum computers.

Security

- How confident are we that our problems are resistant to quantum computers?
  - The most efficient lattice-based cryptosystems use special lattices like ideal lattices.
- How much can we trust our security estimates? How precise are they?

Challenges

- Post-quantum cryptography is a revolution

- There is a lot of exciting work to do

Thank you for your attention...

Any question(s)?