(Quantum) Algorithms for ray class groups and Hilbert class fields

Sean Hallgren

joint with
Kirsten Eisentraeger
Penn State

Quantum Algorithms

- Quantum algorithms for number theoretic problems:
  - Factoring
  - Pell's equation
  - Number fields
    - Unit group
    - Class group
    - Principal ideal problem

- Goal: compute extensions of number fields

Number Field Applications

- Number fields: $\mathbb{Q}(\theta)$
  - Number field sieve
  - Buchmann-Williams key-exchange

- Towers of number fields: $\mathbb{Q}(\theta_1) \subseteq \mathbb{Q}(\theta_2) \subseteq \mathbb{Q}(\theta_3) \subseteq \cdots$
  - Lattice-based crypto
  - Error correcting codes

Number Field Examples

0) $\mathbb{Q}(\theta) = \{\sum_{i=0}^{n-1} a_i\theta^i : a_i \in \mathbb{Q}\}$ $\qquad \theta$ algebraic

1) $\mathbb{Q}$

2) $\mathbb{Q}(\omega_p)$ $\qquad \omega_p = e^{2\pi i/p} \qquad x^p - 1 = 0$

$$a_{p-2}\omega_p^{p-2} + \cdots + a_1\omega_p + a_0 \qquad a_i \in \mathbb{Q}$$

degree $p-1$ $\qquad \sum_{i=0}^{p-1} \omega_p^i = 0$

3) $\mathbb{Q}(\sqrt{d}) \qquad d \in \mathbb{Z}_{>0}$

$$\alpha = a + b\sqrt{d}$$

$$\alpha\overline{\alpha} = (a+b\sqrt{d})(a-b\sqrt{d}) = a^2 - b^2 d$$

Computing Extensions

Output: $\mathbb{Q}(\theta_2)$

Input: $\mathbb{Q}(\theta_1)$

$$\begin{array}{c} \mathbb{Q}(\omega_p, \omega_q) \\ | \\ \mathbb{Q}(\omega_p) \end{array} \qquad \begin{tikzcd} \mathbb{Q}(\theta_1, \omega_p) \arrow[d, no head] \arrow[dr, no head] & \\ \mathbb{Q}(\theta_1) & \mathbb{Q}(\omega_p) \end{tikzcd}$$

where $p - 1 > n$

Hilbert class field of $\mathbb{Q}(\theta_1)$

Maximal abelian unramified extension

$$\begin{array}{c} \mathbb{Q}(\theta_2) \\ | \\ \mathbb{Q}(\theta_1) \end{array}$$

Abelian: Galios group of $\mathbb{Q}(\theta_2)/\mathbb{Q}(\theta_1)$

Unramified: $\mathfrak{p} \cdot \mathcal{O} = \Pi_{\mathfrak{q}} \mathfrak{q}^{e_{\mathfrak{q}}}$

$$\begin{array}{c} \mathfrak{p} \cdot \mathcal{O} \\ \uparrow \\ \mathfrak{p} \end{array}$$

$e_{\mathfrak{q}} = 0, 1 \ \ \forall \mathfrak{q}$

plus real embeddings...

Algorithms

Theorem 1:

Computing the Hilbert class field (a degree 2 subextension) $\boxed{\text{reduces to}}$

Computing:

1) unit group
2) class group
3) factoring ideals
4) computing discrete logs in finite fields

Theorem 2:

computing the ray class group $\boxed{\text{reduces to}}$

Computing:

1) unit group
2) class group
3) principal ideal problem
4) factoring $\mathfrak{m}$
5) computing discrete logs in finite fields

Reductions are efficient: $poly(\log(\Delta))$

Motivation:
Some background on lattice and crypto

Quantum and Crypto

- Quantum can break:
  - RSA
  - Diffie-Hellman
  - Elliptic curve crypto
  - Buchmann-Williams key-exchange
  - Some algebraically homomorphic encr
- Secure against quantum (so far):
  - Lattice-based crypto
  - McEliece
  - MRV proposal based on Hidden Subgroup

8

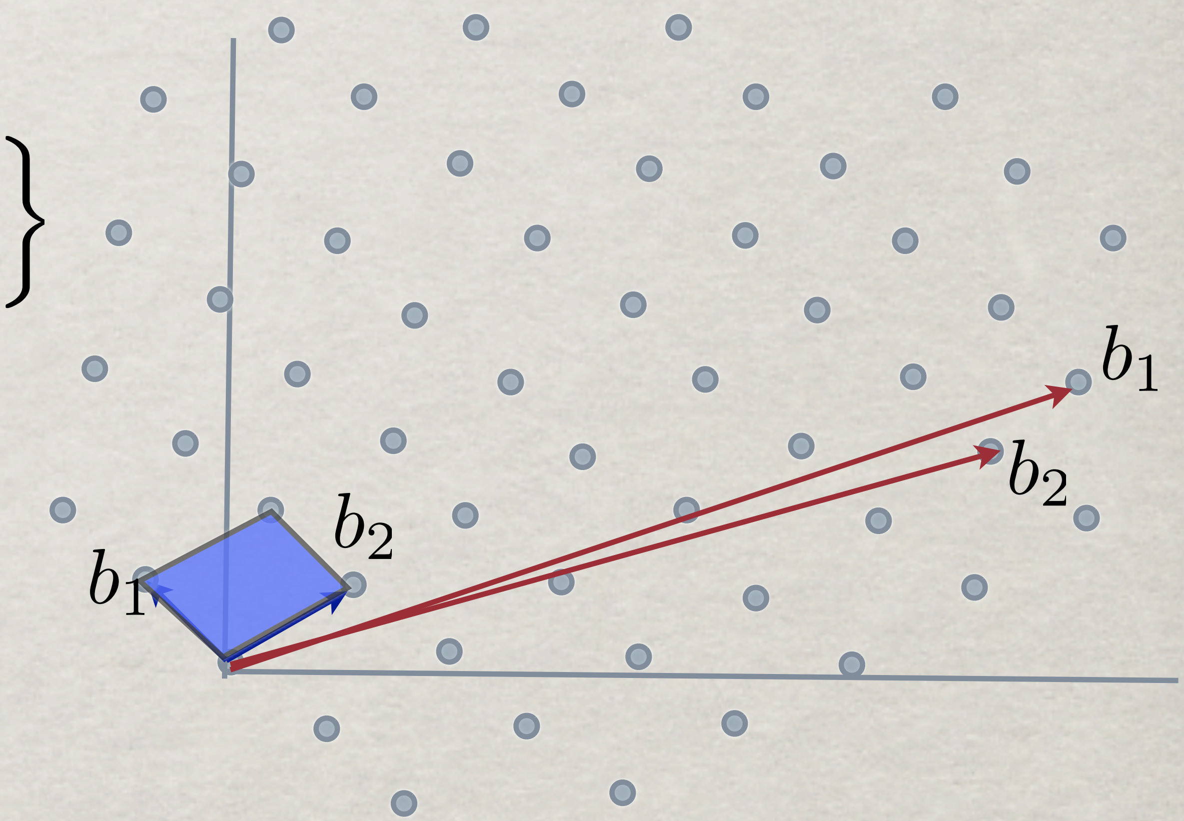
Lattice Based Crypto

- RSA has average-case assumption
  - breaking RSA $\leq$ factoring
- Lattices can provide stronger security:
  - worst case lattice problem
    $\leq$ breaking cryptosystem
- Three directions in lattice-based crypto:
  - Improve worst-case assumption
  - Make more efficient
  - Build more primitives
- Use special lattices

Lattices

- Given $b_1, ..., b_n \in \mathbb{R}^n$

$$L = \left\{ \sum a_i b_i : a_i \in \mathbb{Z} \right\}$$



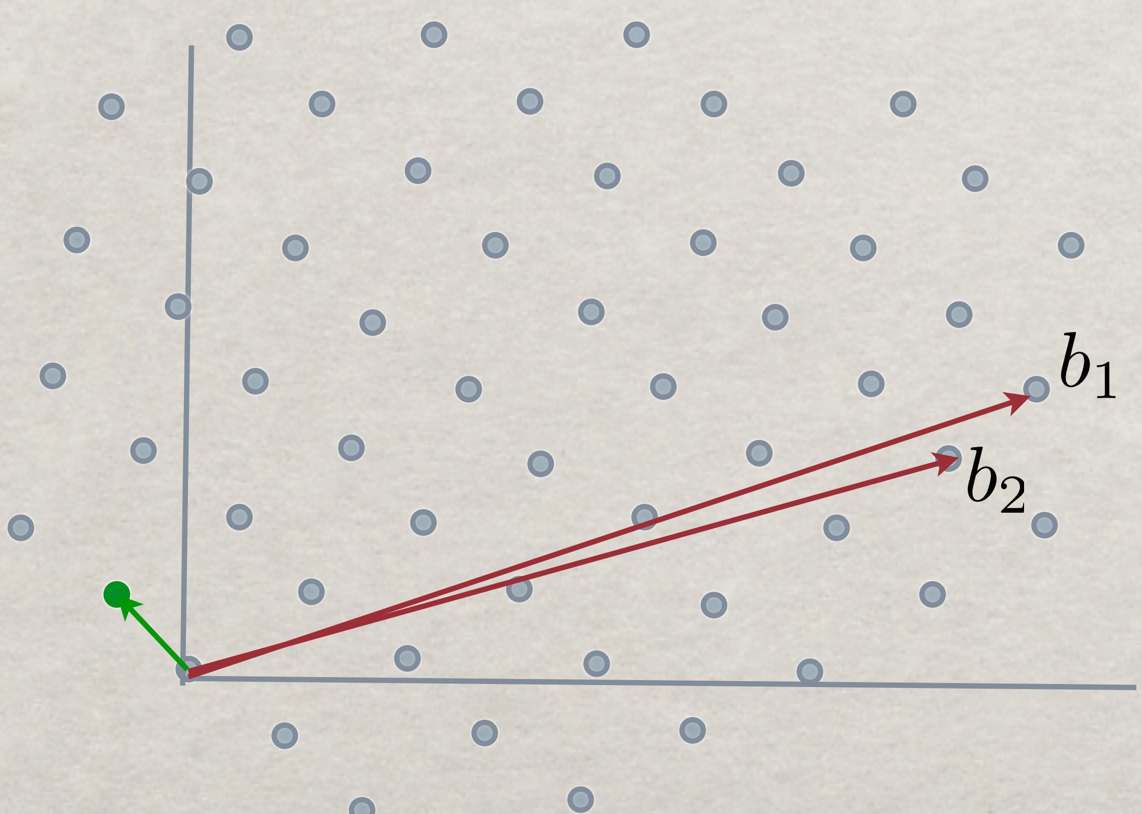
- Infinite number of bases for a lattice

Shortest Vector Problem (SVP)

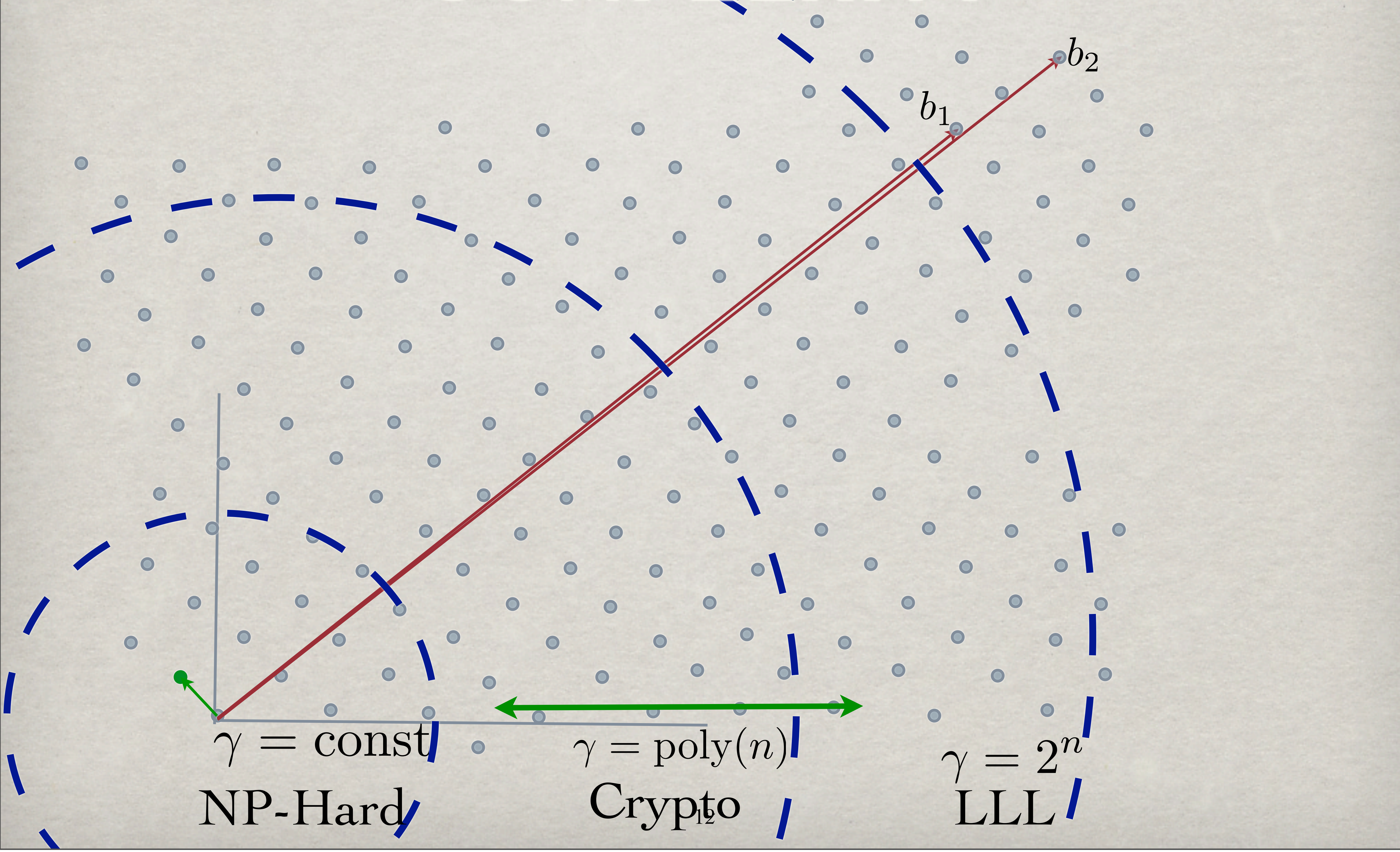
- Given $b_1, ..., b_n \in \mathbb{R}^n$

$$L = \left\{ \sum a_i b_i : a_i \in \mathbb{Z} \right\}$$

- Compute the shortest vector



Approximate-SVP Complexity



One-Way Functions from Cyclic Lattices

- Hash function: rnd $A \in \mathbb{Z}_q^{n \times m}$ $f(y) = Ay \bmod q$
  - Simple, but inefficient in practice
- One-way function: circulant matrix $A \in \mathbb{Z}_q^{n \times m}$

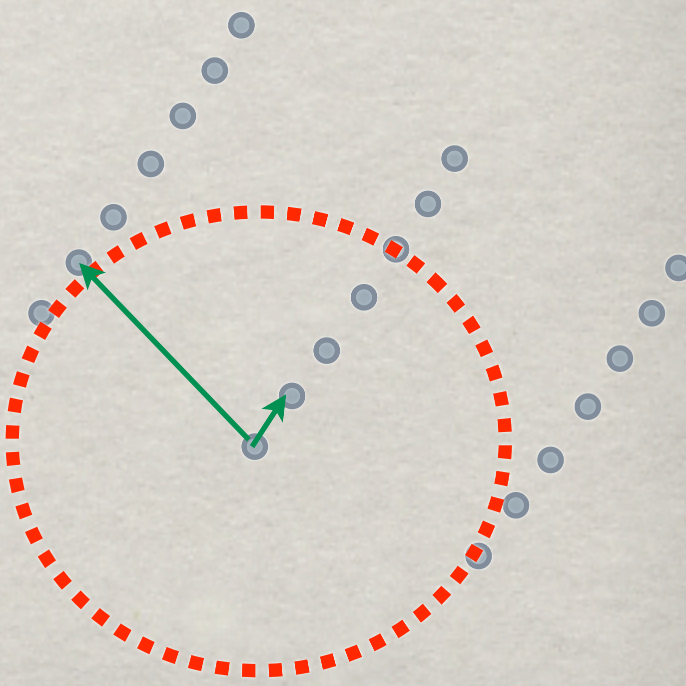
$$A = \left[\begin{array}{ccc|c|ccc} \boxed{a_1 \quad a_2 \quad a_3} & & & & z_1 & z_2 & z_3 \\ a_2 & a_3 & a_1 & \cdots & z_2 & z_3 & z_1 \\ a_3 & a_1 & a_2 & & z_3 & z_1 & z_2 \end{array}\right]$$

  - Worst-case assumption approx-SVP for cyclic lattices, and only for one-way
- Hash function: ideal lattices from $\mathbb{Z}[x]/\langle f(x) \rangle$
  - Worst-case assumption is for ideal lattices.

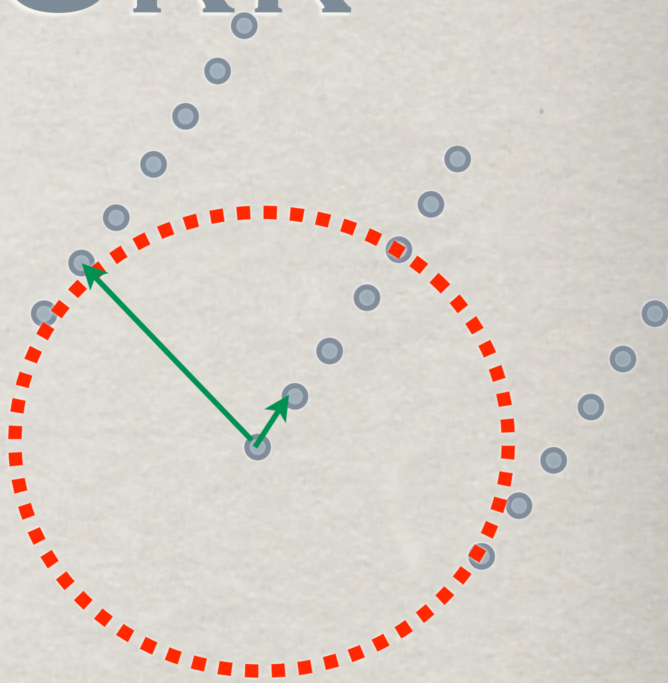
Variations

- Goals:
  - Improve efficiency
    - Want to compete with RSA
  - Reduce approximation factor γ
    - Something between constant and 2^n

- Change the worst-case assumption
  Use special lattices:
  - unique shortest vector
  - ideal lattices



Recent Lattice Work



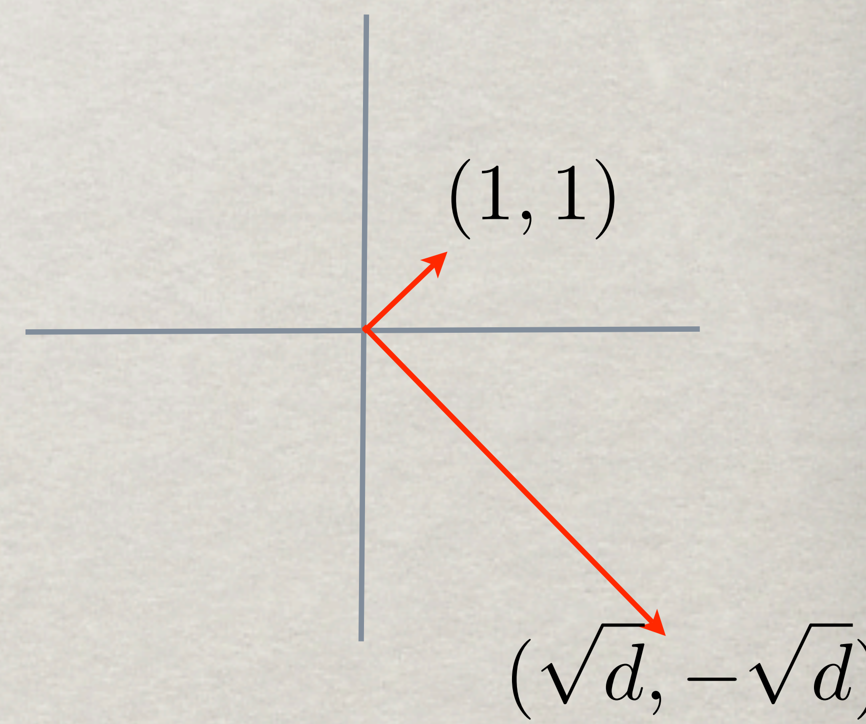
- Ajtai/Dwork97
  - Assume unique-SVP hard
- Regev05: based on SVP
  but the reduction is quantum
  - Assume no quantum alg for SVP
- Peikert08: based on SVP
- Ideal lattices:
  - Micciancio02: more efficient hash function
  - Peikert/Rosen07: improve connection factor from poly(n) to log(n)
    - Assume SVP hard in ideal lattices

Special Lattices: Ideal Lattices

Number field $\mathbb{Q}(\sqrt{d})$ $\quad a + b\sqrt{d}$

Ring of integers $\mathbb{Z}[\sqrt{d}\,]$

Ideals $I_1 \quad I_2 \quad I_3 \quad \cdots$



Ideals map to sublattices

16

Ideal Lattices and Number Fields

$\mathbb{Q}(\theta_1) \longrightarrow L_1, L_2, L_3, ...$

deg = dim

worst-case to average-case reduction (Peikert/Rosen)

$\mathbb{Q}(\omega_5) \longrightarrow L_1 = \{\sum a_i b_i \ : \ a_i \in \mathbb{Z}\}$

Embeddings:

$1, (\omega_5)^1, (\omega_5)^2, (\omega_5)^3$

$1, (\omega_5^2)^1, (\omega_5^2)^2, (\omega_5^2)^3$

$$b_1 = (1, \quad 1, \quad 1, \quad 1)$$

$$b_2 = (\omega_5^1, \omega_5^2, \omega_5^3, \omega_5^4)$$

$$b_3 = (\omega_5^2, \omega_5^4, \omega_5^1, \omega_5^3)$$

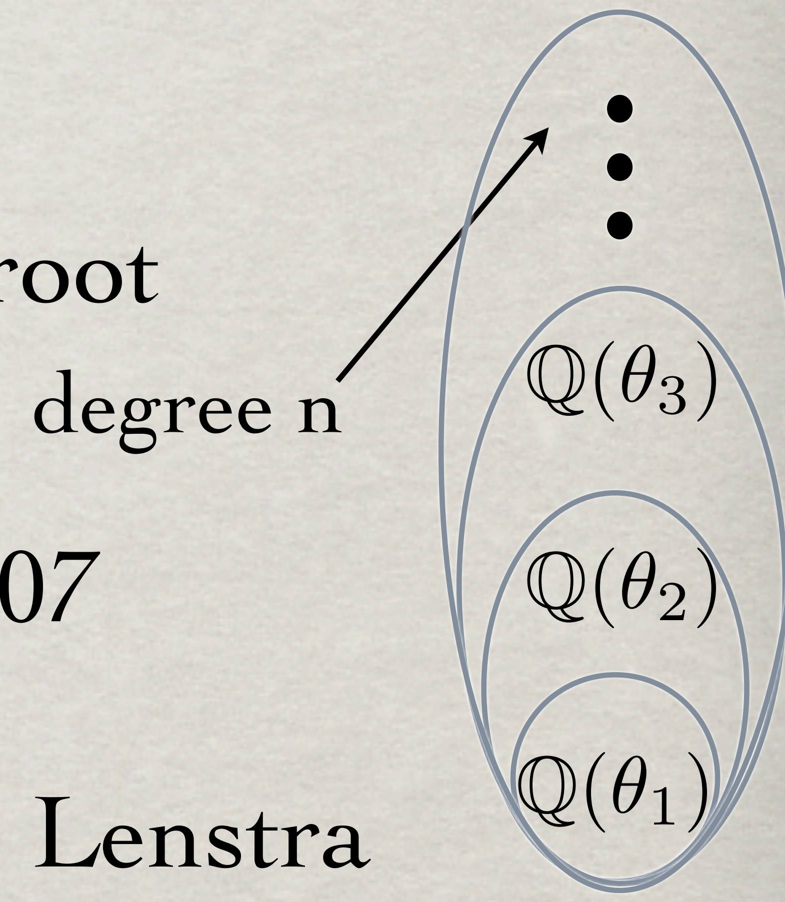
$$b_4 = (\omega_5^3, \omega_5^1, \omega_5^4, \omega_5^2)$$

$L_2, L_3, ...$ Take all sublattices of L_1

Back to computing towers

Computing Number Field Towers

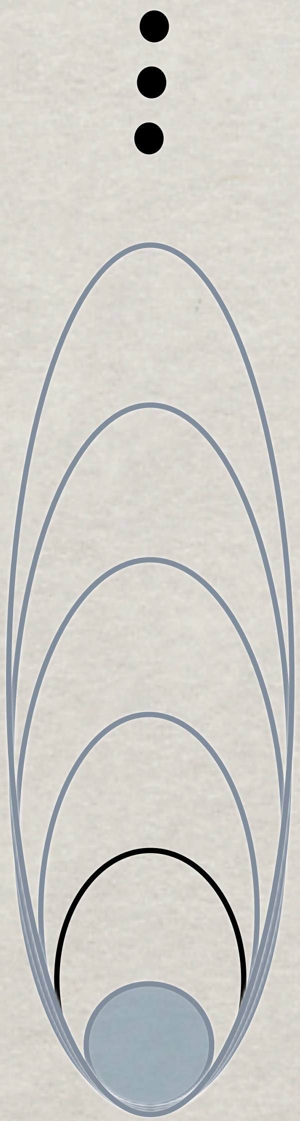
- Input: degree n
  Output: number field with bounded root discriminant $\Delta^{1/n}$



- Lattice-based crypto - Peikert/Rosen07
  Connection factor $\approx \Delta^{1.5/n}\sqrt{\log n}$
- Error correcting codes - Guruswami, Lenstra
  Rate: $R(C) = \cdots - \Delta^{1/n}$
- Existence using Hilbert class fields
  $$\mathbb{Q}(\theta_1) \subseteq \mathbb{Q}(\theta_2) \subseteq \mathbb{Q}(\theta_3) \subseteq \cdots$$
- **Goal**: compute the number fields in the tower

19

Computing Number Fields from Towers



Strategy:
Start with a number field of small degree
Iterate until degree is n:
  Compute the Hilbert class field

Two good base fields:

$$\mathbb{Q}(\sqrt{9699690}) \qquad \mathbb{Q}(\sqrt{-30030})$$

The extension depends on the class group.
  Degree is a problem in the running time.

20

Number Field Problems

Given number field:

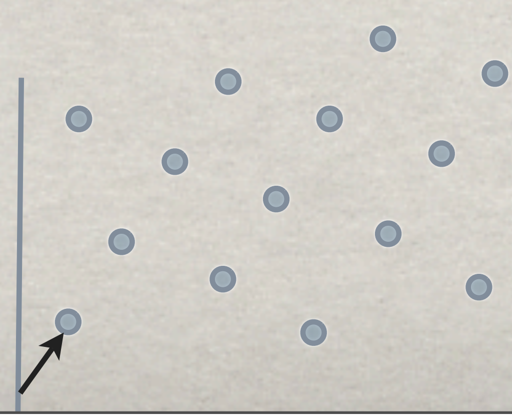
$\mathbb{Q}(\theta)$

Ring of integers $\mathcal{O}$

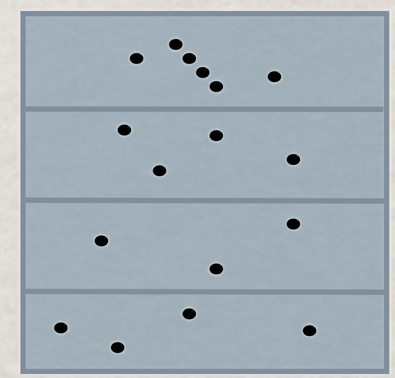
Ideals I_1 I_2 I_3 $\ldots$

Compute:

1) Unit group $\mathcal{O}^*$ = Invertible elements of $\mathcal{O}$



2) Class group = Ideals mod Principal ideals



3) Principal ideal problem

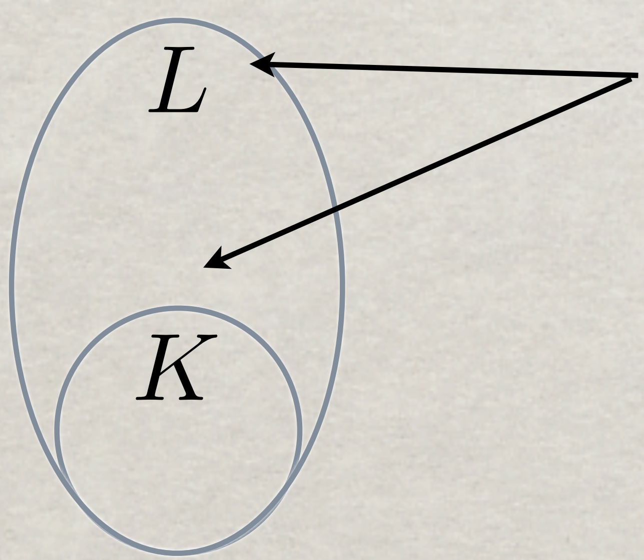
$$\alpha\mathcal{O} \mapsto \alpha$$

Quantum algorithms for constant degree cases

21

Hilbert Class Field L of K

- Hilbert class field
  - maximal unramified abelian extension
  - Constant root discriminant $\Delta^{1/n}$



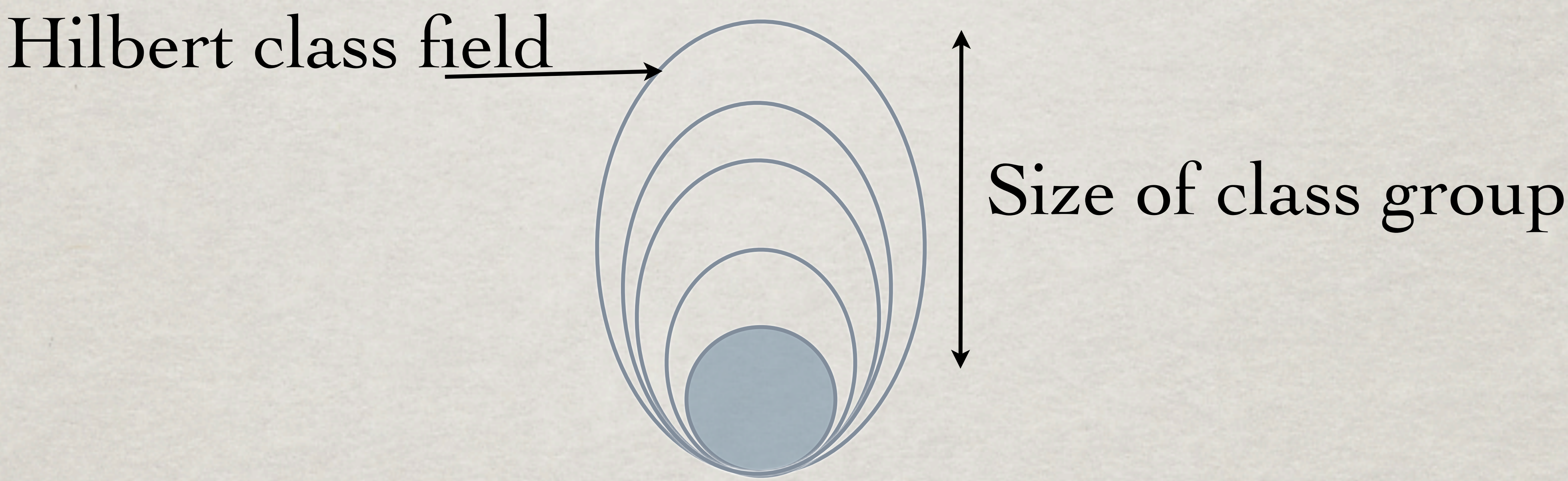
Described by class group of K

Degree of L/K
= size of class group of K

1) Could be trivial: no extension, L=K
2) Could be exponential size: can't write down

22

Computing Hilbert Class Fields



Theorem 1:

Efficient quantum algorithm for degree two extensions in the Hilbert class field

(Still has constant root discriminant)

23

Computing Hilbert Class Fields

- Ingredients:
  - Change to compact representations
  - Virtual units
  - The group $(O/m)^*$
  - Ideal factorization
- We show these efficiently reduce to unit group, class group, etc.

24

Ideal Factorization

Given $I \subseteq \mathcal{O}$ compute $I = \mathfrak{p}_1^{e_1} \cdots \mathfrak{p}_k^{e_k}$

Algorithm:

1. Factor the norm $N(I) = p_1^{e_1} \cdots p_k^{e_k}$
2. Compute the set of prime ideals $\mathfrak{p}$ above each prime integer p
3. Compute valuations of each prime

$$\begin{tikzcd}[column sep=small] \mathfrak{p}_1 \arrow[dr, no head] & \cdots & \mathfrak{p}_\ell \arrow[dl, no head] \\ & p & \end{tikzcd}$$

We show steps 2 and 3 are efficient.

Computing Primes Above p: Easy Case

$K = \mathbb{Q}(\theta)$

Easy case: $p \not| [\mathcal{O}_K : \mathbb{Z}[\theta]]$

f = minimal polynomial of θ

Factor $f(x) = \Pi_i f_i(x)^{e_i}$ over $\mathbb{F}_p$

The primes above: p

$$\mathfrak{p}_i = p\mathcal{O}_K + f_i(\theta)\mathcal{O}_K$$

Computing Primes Above p: Hard Case

p-Radical: $I_p = \{x \in \mathcal{O}_K : x^m \in p\mathcal{O}_K \text{ for some } m \in \mathbb{Z}^+\}$

Claim: $I_p = \Pi_i \mathfrak{p}_i$ product over primes $\mathfrak{p}$ above p

$$\mathcal{O}_K/I_p \cong \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_k \qquad \text{(CRT)}$$

Finites fields

1) Compute I_p

2) Given $I = \mathfrak{p}_1 \cdot \mathfrak{p}_2 \cdots \mathfrak{p}_k$ distinct primes over p

Compute $\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_k$

Computing Primes Above p: Hard Case

1) Computing $I_p = \Pi_i \mathfrak{p}_i$

Compute $\mathbb{F}_p$ basis of $I_p/p\mathcal{O}_K$

Compute $\ker(x \mapsto x^q) = I_p/p\mathcal{O}_K$

the radical of $\mathcal{O}_K/p\mathcal{O}_K$

Compute I_p

Use generators of $I_p/p\mathcal{O}_K$ and $p\mathcal{O}_K$

Computing Primes Above p: Hard Case

2) Given $I = \mathfrak{p}_1 \cdot \mathfrak{p}_2 \cdots \mathfrak{p}_k$ distinct primes over

Compute $\mathfrak{p}_1, \mathfrak{p}_2, \ldots, \mathfrak{p}_k$

Compute an idempotent $e \in \mathcal{O}_K/I$

$e \neq 0, 1$

$(1,0)^2 = (1,0)$

$$e(1-e) = e - e^2 = 0 \in \mathcal{O}_K/I$$

Compute

$$H_1 = I + e\mathcal{O}_K$$

$$H_2 = I + (1-e)\mathcal{O}_K$$

$I = H_1 H_2$ is a nontrivial factorization

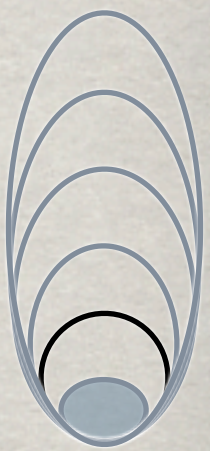
$$I^2 + eI + (1-e)I + e(1-e)\mathcal{O}_K \subseteq I$$

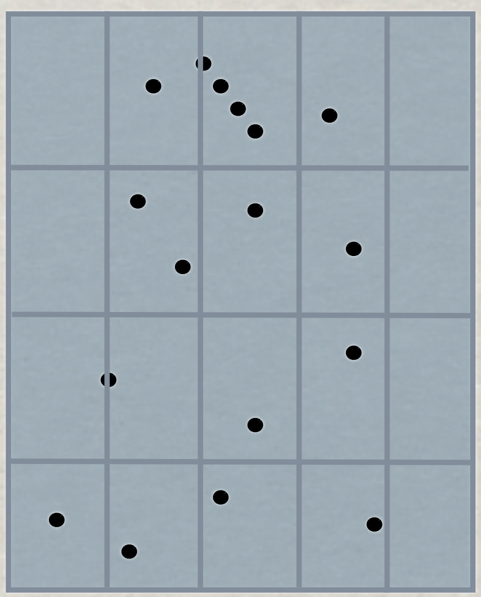
$$I \subseteq eI + (1-e)I : \quad e\alpha + (1-e)\alpha = \alpha \in I$$

Summary

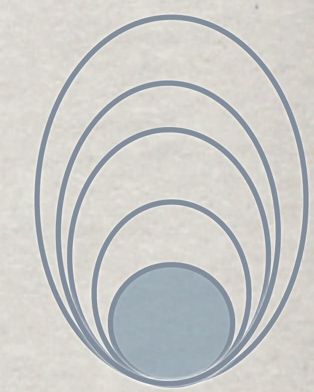
- Two basic objects in class field theory that also appear in apps in computer science.

- We gave efficient quantum algorithms for:
  1. Degree two extensions in the Hilbert class field
  2. The ray class group



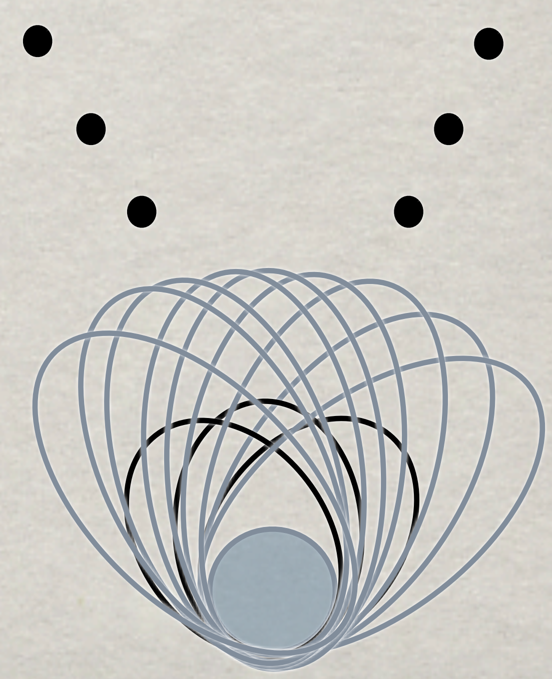


Compute Towers?



- Goal: compute towers
  - Compute larger subfields of Hilbert class fields
  - Compute multiple steps in a tower
  - Compute ray class field towers
    - **Theorem:** Q. alg for the ray class group

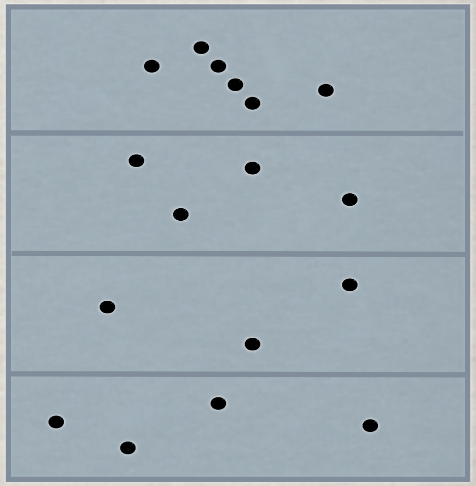
$$U \xrightarrow{\rho} (\mathcal{O}_K/\mathfrak{m})^* \xrightarrow{\psi} \mathrm{Cl}_{\mathfrak{m}} \xrightarrow{\phi} \mathrm{Cl} \rightarrow 1$$



Open Problem: Arbitrary Degree

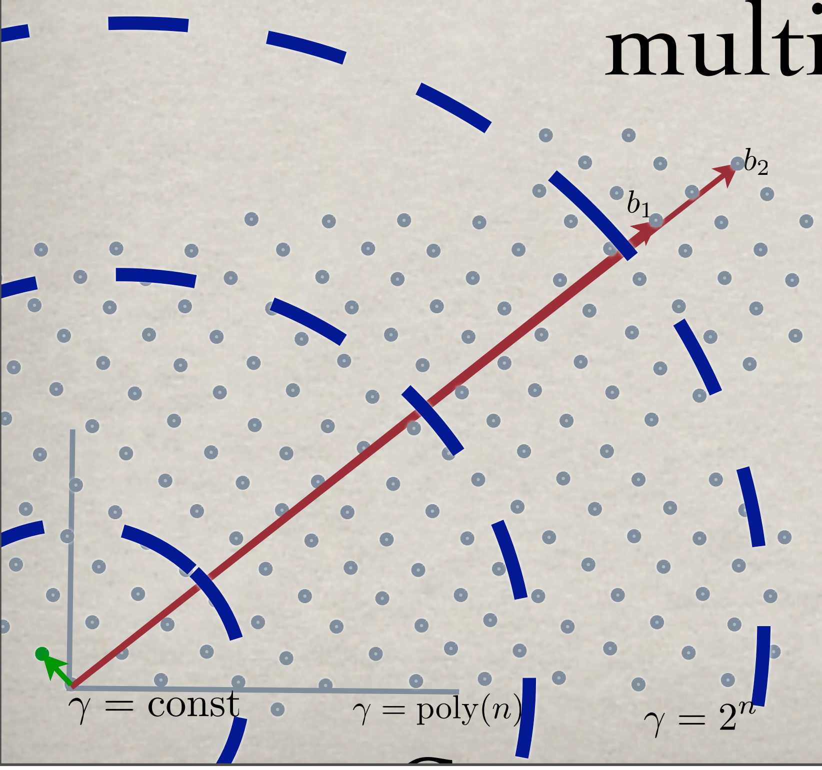
- Hilbert class field iterations require class group computations (at least)
  - SVP in ideal lattices must be solved
- Use superpositions to bypass this?
- Rework definitions so SVP not necessary?

Ideals



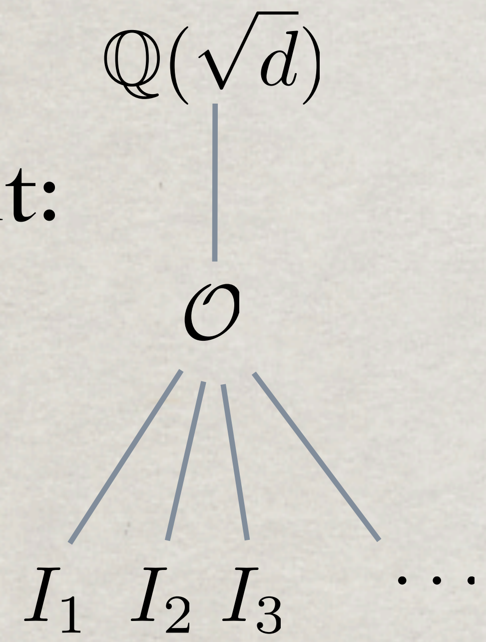
Open Problem

- Quantum algorithm for SVP in ideal lattices?
  - Two extra features:
    - For constant root discriminant, the length of the shortest vector can be efficiently approximated.
    - The lattice is also an ideal: closed under multiplication.

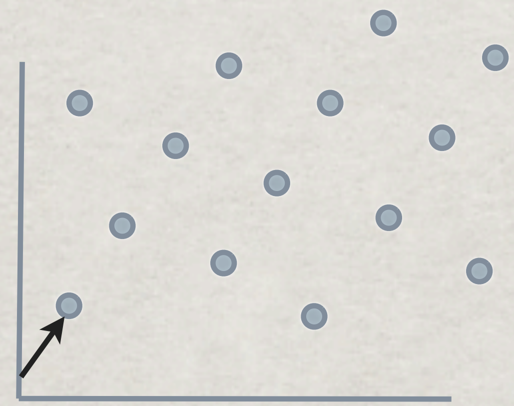


Main Problems

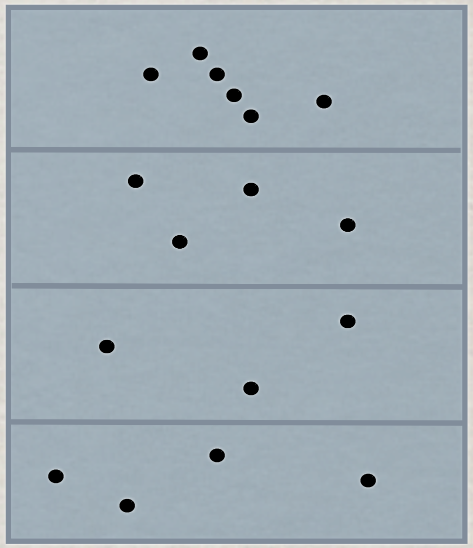
Input:



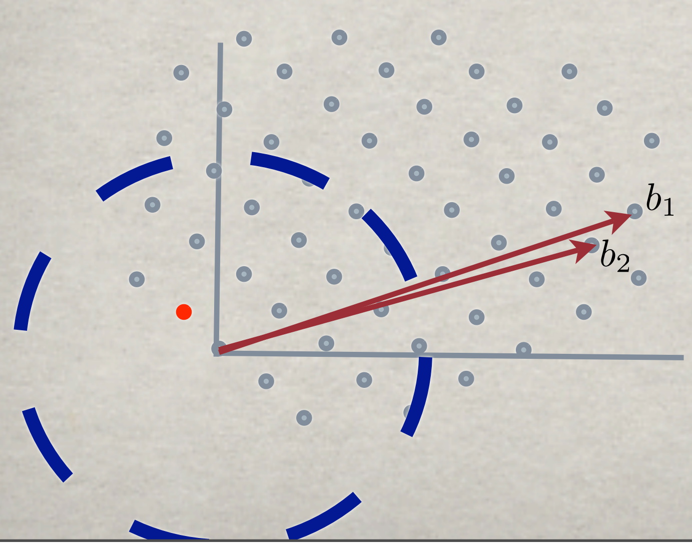
Unit group



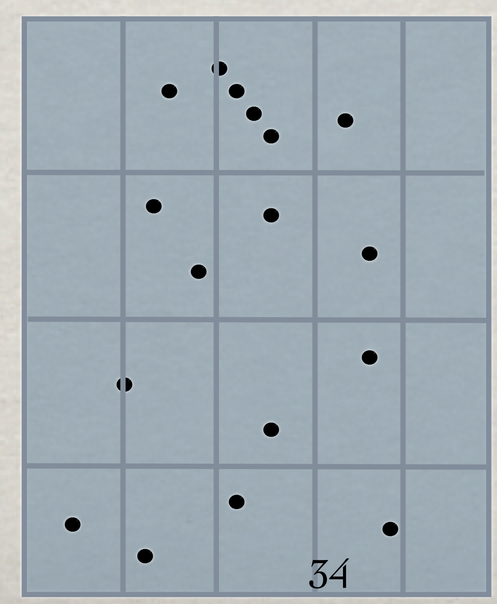
Principal ideal problem



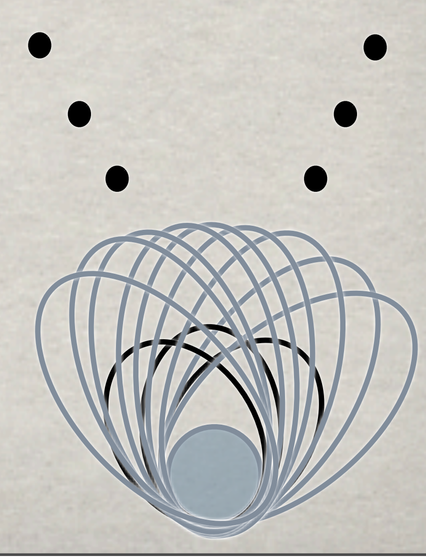
Shortest lattice vector



Ray Class group



Hilbert/Ray Class Field Tower



34