Approx-SVP in Ideal Lattices with Pre-Processing

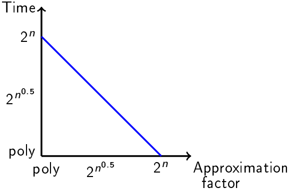
Alice Pellet-Mary, Guillaume Hanrot and Damien Stehlé

ENS de Lyon

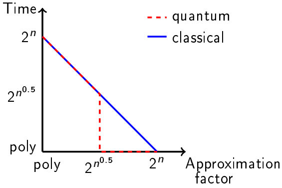
Eurocrypt
May 21, 2019

What is this talk about

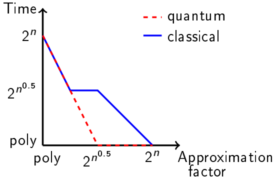
Time/Approximation trade-offs for SVP in ideal lattices:



BKZ algorithm [Sch87]



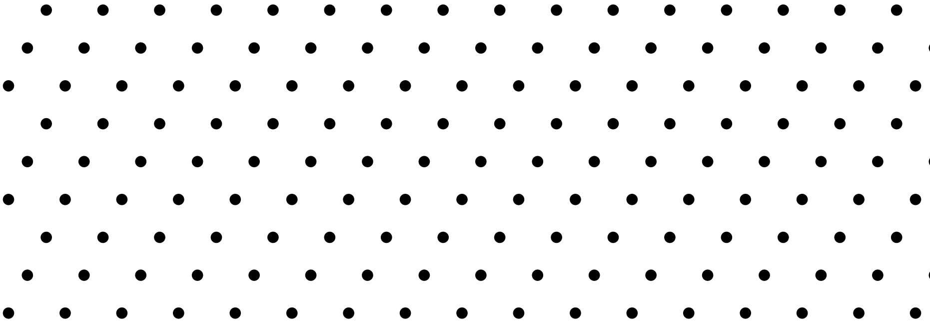
[CDW17]



This work
(with $2^{O(n)}$ pre-processing)

(Figures are for prime power cyclotomic fields)

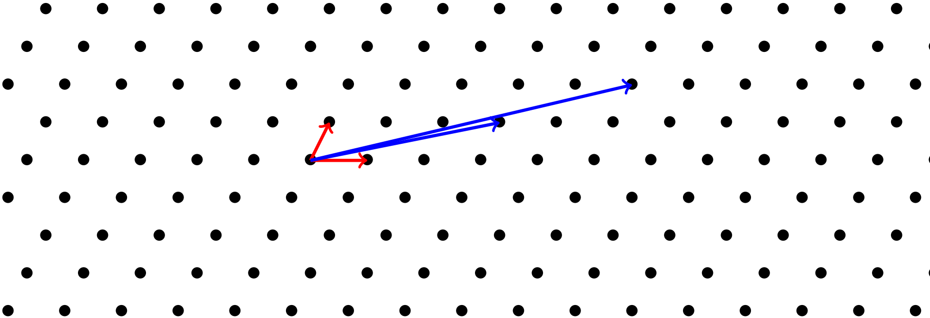
Lattices



Lattice

A lattice L is a discrete 'vector space' over $\mathbb{Z}$.

Lattices

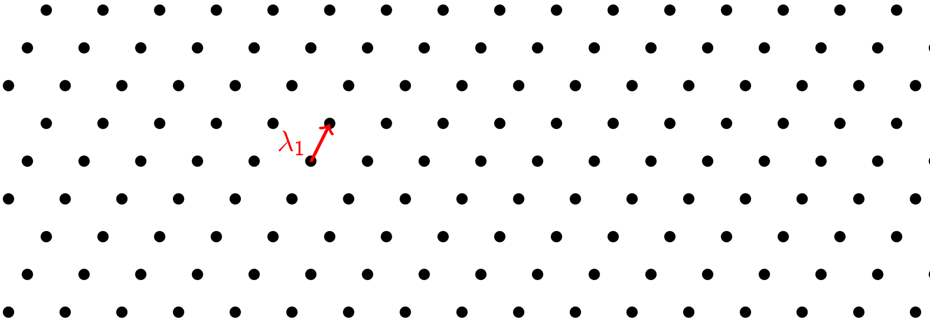


Lattice

A lattice L is a discrete 'vector space' over $\mathbb{Z}$.
A basis of L is an invertible matrix B such that $L = \{Bx \mid x \in \mathbb{Z}^n\}$.

$\begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix}$ and $\begin{pmatrix} 17 & 10 \\ 4 & 2 \end{pmatrix}$ are two bases of the above lattice.

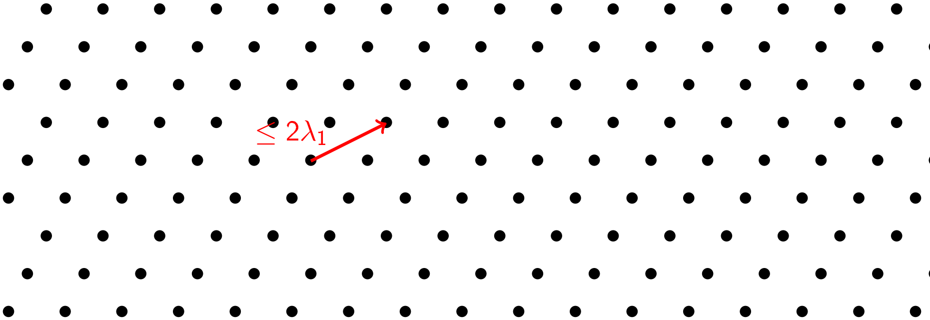
Lattices



Shortest Vector Problem (SVP)

Find a shortest (in Euclidean norm) non-zero vector.
Its Euclidean norm is denoted λ_1.

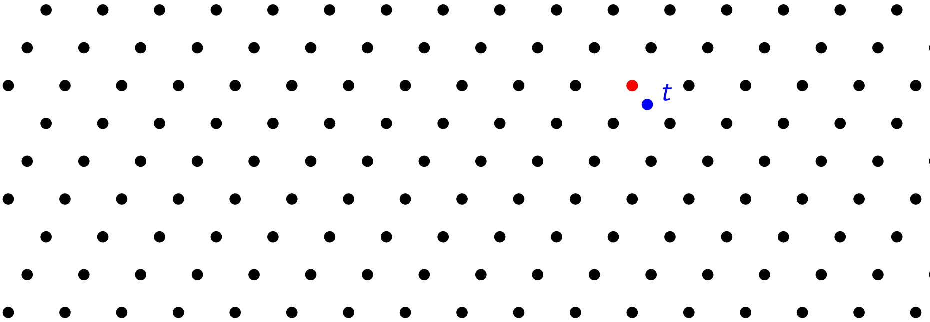
Lattices



Approximate Shortest Vector Problem (approx-SVP)

Find a short (in Euclidean norm) non-zero vector.
(e.g. of norm $\leq 2\lambda_1$).

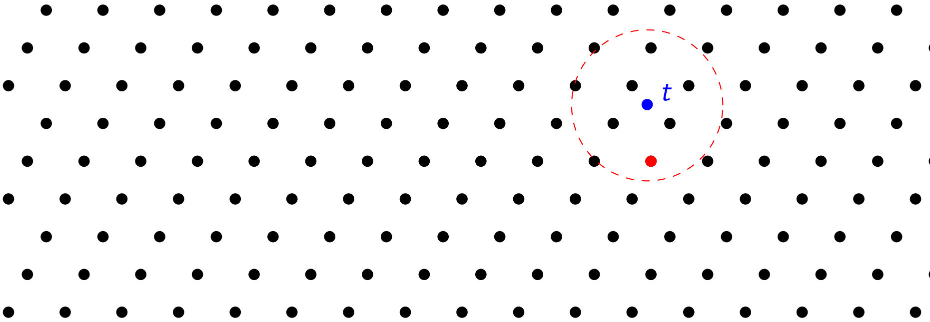
Lattices



Closest Vector Problem (CVP)

Given a target point t, find a point of the lattice closest to t.

Lattices



Approximate Closest Vector Problem (approx-CVP)

Given a target point t, find a point of the lattice close to t.

Complexity of SVP/CVP

Applications

Approx-SVP and approx-CVP in generic lattices are conjectured to be hard to solve both quantumly and classically $\Rightarrow$ used in cryptography

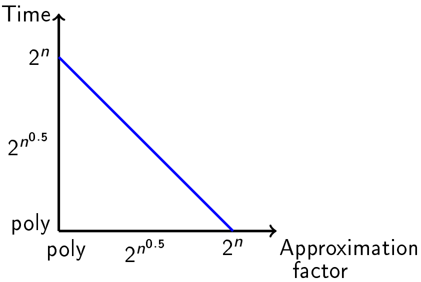
[Sch87] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical computer science.

Complexity of SVP/CVP

Applications

Approx-SVP and approx-CVP in generic lattices are conjectured to be hard to solve both quantumly and classically $\Rightarrow$ used in cryptography

Best Time/Approx trade-off for arbitrary lattices: BKZ algorithm [Sch87]



[Sch87] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical computer science.

Structured lattices

Improve efficiency of lattice-based crypto using structured lattices.
$\Rightarrow$ E.g. ideal lattices = ideals in the ring of integers of a number field

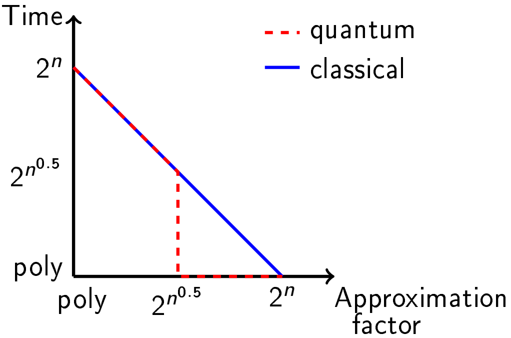
Structured lattices

Improve efficiency of lattice-based crypto using structured lattices.
$\Rightarrow$ E.g. ideal lattices = ideals in the ring of integers of a number field

Is approx-SVP still hard when restricted to ideal lattices?

SVP in ideal lattices

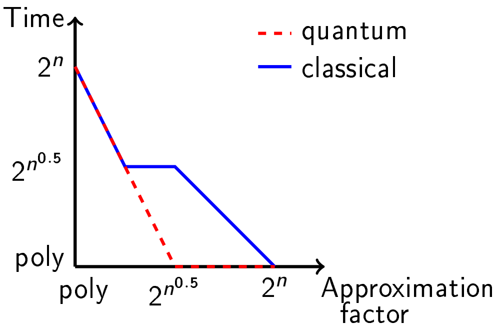
[CDW17]: Better than BKZ in the quantum setting



- Heuristic
- For prime power cyclotomic fields

[CDW17] R. Cramer, L. Ducas, B. Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP, Eurocrypt.

This work



(Figure for prime power cyclotomic fields)

- Heuristic
- Pre-processing $2^{O(n)}$, independent of the choice of the ideal
- All number fields (trade-offs differ slightly)

Impact

- Approx-SVP in ideal lattices might be easier than in generic lattices

Impact

- Approx-SVP in ideal lattices might be easier than in generic lattices
- No concrete impact/attack against crypto schemes
  - exponential pre-processing

Impact

- Approx-SVP in ideal lattices might be easier than in generic lattices
- No concrete impact/attack against crypto schemes
  - exponential pre-processing
  - very few schemes based in ideal-SVP [Gen09,GGH13]

schemes ⟶ RLWE ⟶ ideal SVP

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices, STOC.

[GGH13] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices, Eurocrypt.

Impact

- Approx-SVP in ideal lattices might be easier than in generic lattices
- No concrete impact/attack against crypto schemes
  - exponential pre-processing
  - very few schemes based in ideal-SVP [Gen09,GGH13]

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices, STOC.

[GGH13] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices, Eurocrypt.

Outline of the talk

1. Definitions and objective
2. The CDPR algorithm
3. This work

First definitions

Notation

$R = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$ (for simplicity)

First definitions

Notation

$R = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$ (for simplicity)

- Units: $R^\times = \{a \in R \mid \exists b \in R, ab = 1\}$
  - e.g. $\mathbb{Z}^\times = \{-1, 1\}$

First definitions

Notation

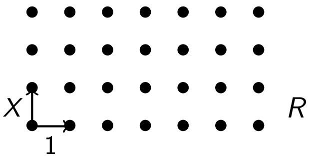
$R = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$ (for simplicity)

- Units: $R^\times = \{a \in R \mid \exists b \in R, ab = 1\}$
  - e.g. $\mathbb{Z}^\times = \{-1, 1\}$
- Principal ideals: $\langle g \rangle = \{gr \mid r \in R\}$ (i.e. all multiples of g)
  - e.g. $\langle 2 \rangle = \{\text{even numbers}\}$ in $\mathbb{Z}$
  - g is called a generator of $\langle g \rangle$
  - The generators of $\langle g \rangle$ are exactly the ug for $u \in R^\times$

Why is $\langle g \rangle$ a lattice?

$R \simeq \mathbb{Z}^n$

$$R = \mathbb{Z}[X]/(X^n + 1) \;\to\; \mathbb{Z}^n$$
$$r = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1} \;\mapsto\; (r_0, r_1, \ldots, r_{n-1})$$

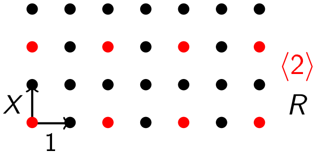


Why is $\langle g \rangle$ a lattice?

$R \simeq \mathbb{Z}^n$

$$R = \mathbb{Z}[X]/(X^n+1) \;\rightarrow\; \mathbb{Z}^n$$
$$r = r_0 + r_1 X + \cdots + r_{n-1}X^{n-1} \;\mapsto\; (r_0, r_1, \ldots, r_{n-1})$$

$\langle g \rangle \subseteq R \simeq \mathbb{Z}^n$ + stable by '+' and '-' $\Rightarrow$ lattice



Objective of this talk

Objective

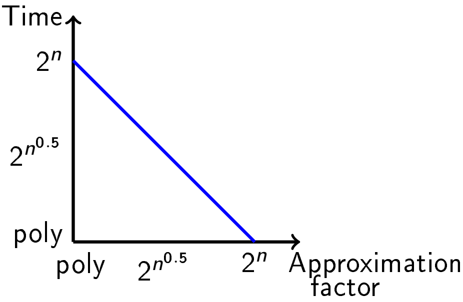
Given a basis of a principal ideal $\langle g \rangle$ and $\alpha \in (0, 1]$,
Find $r \in \langle g \rangle \setminus \{0\}$ such that $\|r\| \leq 2^{\widetilde{O}(n^\alpha)} \cdot \lambda_1$.

Objective of this talk

Objective

Given a basis of a principal ideal $\langle g \rangle$ and $\alpha \in (0,1]$,
Find $r \in \langle g \rangle \setminus \{0\}$ such that $\|r\| \leq 2^{\widetilde{O}(n^\alpha)} \cdot \lambda_1$.

BKZ algorithm can do it in time $2^{O(n^{1-\alpha})}$, can we do better?



The CDPR algorithm (on ideas of [RBV04,CGS14,Ber14])

[CDPR16] R. Cramer, L. Ducas, C. Peikert and O. Regev. Recovering Short Generators of Principal Ideals in Cyclotomic Rings. Eurocrypt.

[RBV04]: G. Rekaya, J.-C. Belfiore, and E. Viterbo. A very efficient lattice reduction tool on fast fading channels. ISITA.

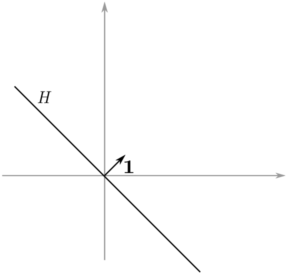
[CGS14]: P. Campbell, M. Groves, and D. Shepherd. Soliloquy: A cautionary tale.

[Ber14]: D. J. Bernstein. A subfield-logarithm attack against ideal lattices: Computational algebraic number theory tackles lattice based cryptography. The cr.yp.to blog.

The Log space

Log : $R \to \mathbb{R}^n$ (somehow generalising log to R)

Let $\mathbf{1} = (1, \cdots, 1)$ and $H = \mathbf{1}^{\perp}$.



The Log space

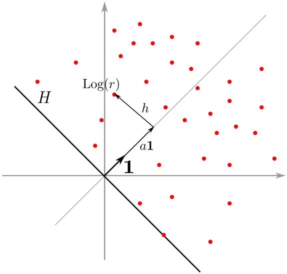
Log : $R \to \mathbb{R}^n$ (somehow generalising log to R)

Let $\mathbf{1} = (1, \cdots, 1)$ and $H = \mathbf{1}^\perp$.

Properties

$\text{Log}\, r = h + a\mathbf{1}$, with $h \in H$

- $a \geq 0$



The Log space

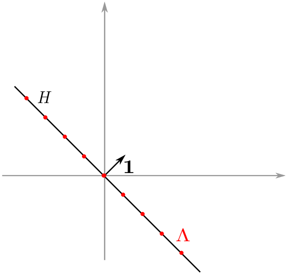
$\text{Log} : R \to \mathbb{R}^n$ (somehow generalising log to R)

Let $\mathbf{1} = (1, \cdots, 1)$ and $H = \mathbf{1}^\perp$.

Properties

$\text{Log}\, r = h + a\mathbf{1}$, with $h \in H$

- $a \geq 0$
- $a = 0$ iff r is a unit
- $\Lambda := \text{Log}(R^\times)$ is a lattice



The Log space

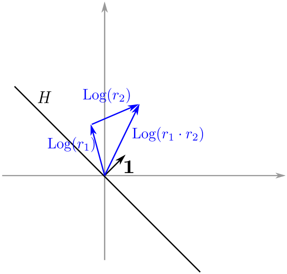
Log : $R \to \mathbb{R}^n$ (somehow generalising log to R)

Let $\mathbf{1} = (1, \cdots, 1)$ and $H = \mathbf{1}^\perp$.

Properties

$\text{Log}\, r = h + a\mathbf{1}$, with $h \in H$

- $a \geq 0$
- $a = 0$ iff r is a unit
- $\Lambda := \text{Log}(R^\times)$ is a lattice
- $\text{Log}(r_1 \cdot r_2) = \text{Log}(r_1) + \text{Log}(r_2)$



The Log space

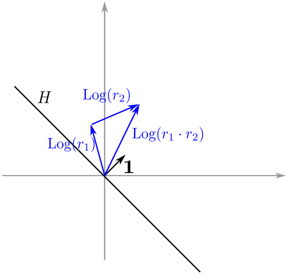
$\mathrm{Log} : R \to \mathbb{R}^n$ (somehow generalising log to R)

Let $\mathbf{1} = (1, \cdots, 1)$ and $H = \mathbf{1}^\perp$.

Properties

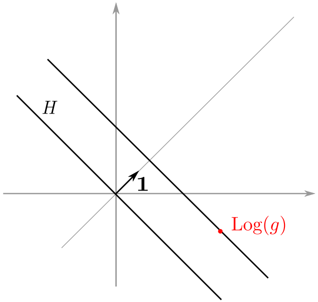
$\mathrm{Log}\, r = h + a\mathbf{1}$, with $h \in H$

- $a \geq 0$
- $a = 0$ iff r is a unit
- $\Lambda := \mathrm{Log}(R^\times)$ is a lattice
- $\mathrm{Log}(r_1 \cdot r_2) = \mathrm{Log}(r_1) + \mathrm{Log}(r_2)$
- $\|r\| \simeq 2^{\|\mathrm{Log}\, r\|_\infty}$



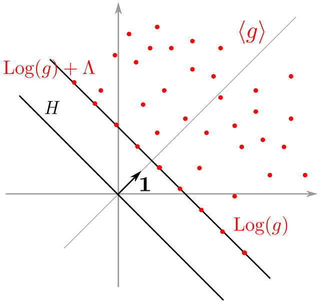
The CDPR algorithm

What does $\mathrm{Log}\langle g \rangle$ look like?



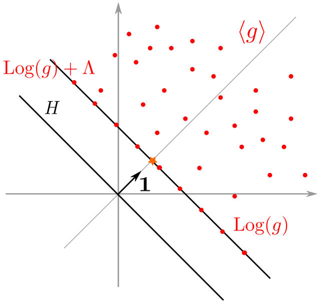
The CDPR algorithm

What does $\text{Log}\langle g \rangle$ look like?



The CDPR algorithm

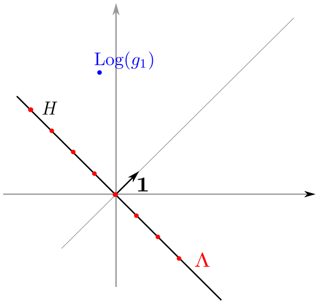
What does $\text{Log}\langle g \rangle$ look like?



The CDPR algorithm

The CDPR algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$



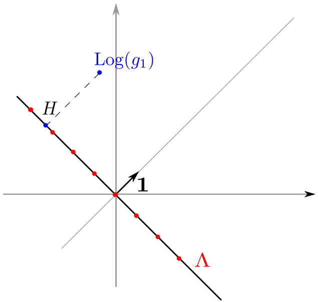
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

The CDPR algorithm

The CDPR algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$
- Solve CVP in Λ



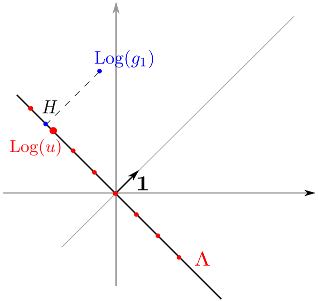
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

The CDPR algorithm

The CDPR algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$
- Solve CVP in Λ



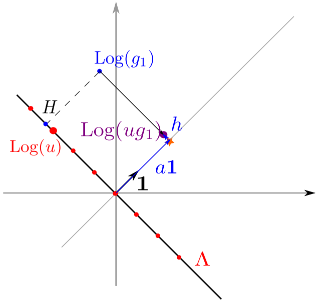
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

The CDPR algorithm

The CDPR algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$
- Solve CVP in Λ



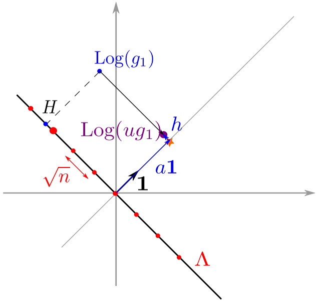
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

The CDPR algorithm

The CDPR algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$
- Solve CVP in Λ
  - Good basis of Λ
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$



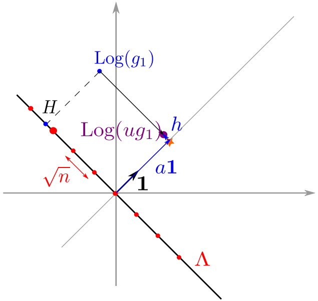
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

The CDPR algorithm

The CDPR algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$
- Solve CVP in Λ
  - Good basis of Λ
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$



$$\|ug_1\| \leq 2^{\widetilde{O}(\sqrt{n})} \cdot \lambda_1$$

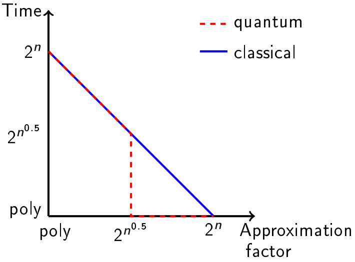
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

The CDPR algorithm

The CDPR algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$
- Solve CVP in Λ
  - Good basis of Λ
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$



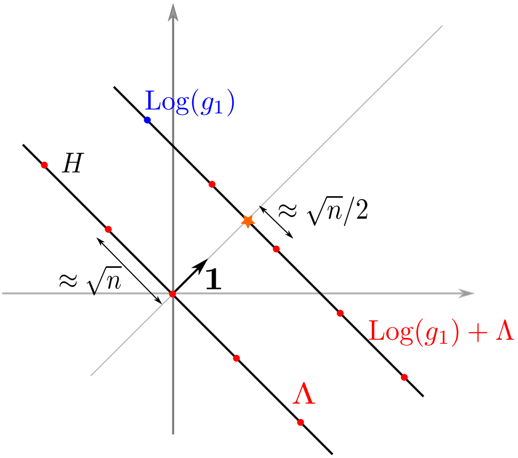
$$\|ug_1\| \leq 2^{\widetilde{O}(\sqrt{n})} \cdot \lambda_1$$

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

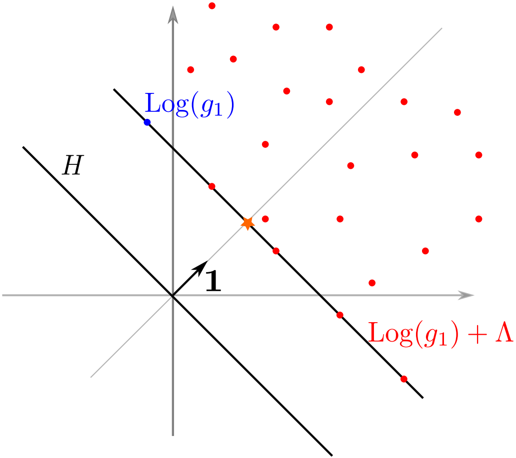
This work

Idea

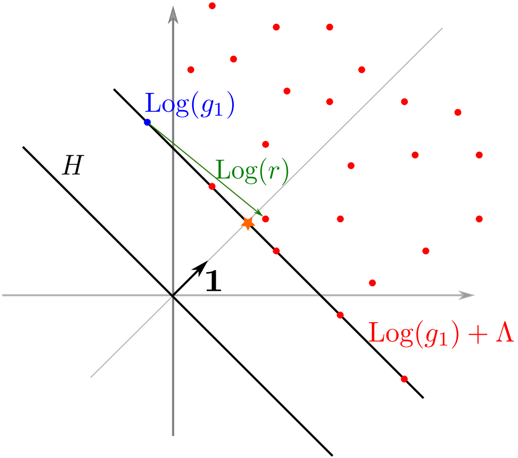


$\mathrm{Log}(g_1)$
H
$\approx \sqrt{n}/2$
$\approx \sqrt{n}$
$\mathbf{1}$
$\mathrm{Log}(g_1) + \Lambda$
Λ

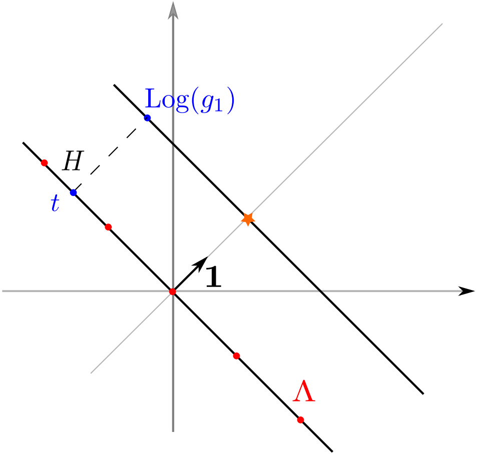
Idea



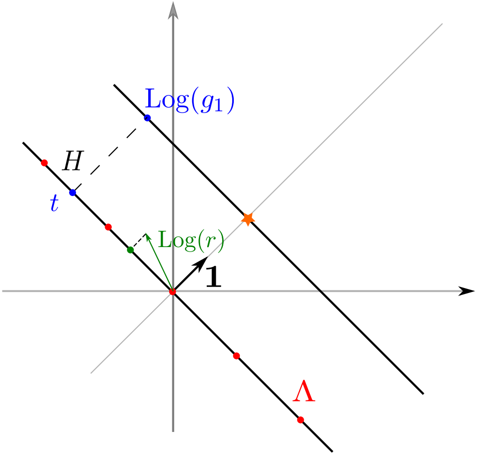
Idea



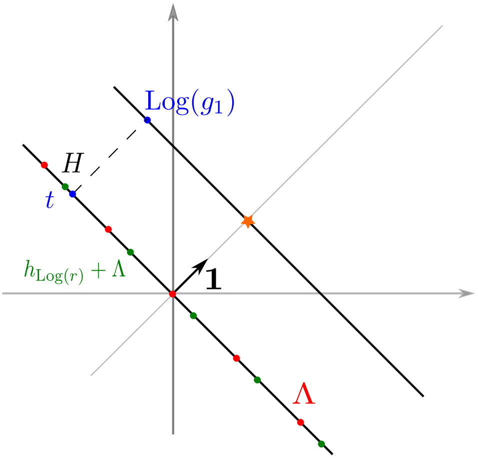
Idea



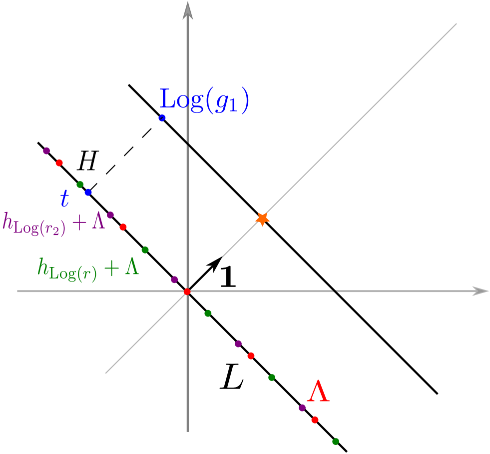
Idea



Idea



Idea



Log(g_1)
H
t
$h_{\mathrm{Log}(r_2)} + \Lambda$
$h_{\mathrm{Log}(r)} + \Lambda$
$\mathbf{1}$
L
Λ

The lattice L

$$L = \left(\begin{array}{c|cccc} \Lambda & \multicolumn{4}{c}{h_{\mathrm{Log}(r_1)}, \cdots, h_{\mathrm{Log}(r_\nu)}} \\ \hline & 1 & & & \\ 0 & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{array}\right) \qquad t = \left(\begin{array}{c} h_{Log(g_1)} \\ \hline \\ 0 \\ \\ \end{array}\right)$$

The lattice L

$$L = \left(\begin{array}{c|c} \Lambda & h_{\mathrm{Log}(r_1)}, \cdots, h_{\mathrm{Log}(r_\nu)} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \end{array}\right) \qquad t = \left(\begin{array}{c} h_{Log(g_1)} \\ \hline 0 \end{array}\right)$$

Heuristic

For some $\nu = \widetilde{O}(n)$, the covering radius of L satisfies $\mu(L) = O(1)$.

$=$ for all target t, there exists $s \in L$ such that $\|t - s\| = O(1)$

How to solve CVP in L?

CDPR	This work
Good basis of Λ	No good basis of L known

How to solve CVP in L?

CDPR	This work
Good basis of Λ	No good basis of L known

Key observation

L does not depend on $\langle g \rangle$

How to solve CVP in L?

CDPR	This work
Good basis of Λ	No good basis of L known

Key observation

L does not depend on $\langle g \rangle$ $\quad \Rightarrow$ Pre-processing on L

How to solve CVP in L?

CDPR	This work
Good basis of Λ	No good basis of L known

Key observation

L does not depend on $\langle g \rangle$ $\Rightarrow$ Pre-processing on L

[Laa16,DLW19,Ste19]:

- Find $s \in L$ such that $\|s - t\| = \widetilde{O}(n^{\alpha})$
- Time:
  - $2^{\widetilde{O}(n^{1-2\alpha})}$ (query)
  - $+\ 2^{O(n)}$ (pre-processing)

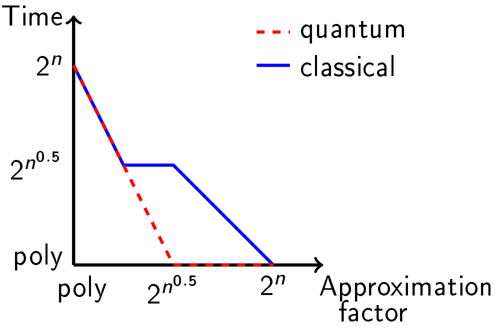
[Laa16] T. Laarhoven. Finding closest lattice vectors using approximate Voronoi cells. SAC.

[DLW19]: E. Doulgerakis, T. Laarhoven, and B. de Weger. Finding closest lattice vectors using approximate Voronoi cells. PQCRYPTO.

[Ste19]: N. Stephens-Davidowitz. A time-distance trade-off for GDD with preprocessing – instantiating the DLW heuristic. ArXiv.

Conclusion

Approximation	Query time	Pre-processing
$2^{\widetilde{O}(n^\alpha)}$	$2^{\widetilde{O}(n^{1-2\alpha})} + (\mathrm{poly}(n)$ or $2^{\widetilde{O}(\sqrt{n})})$	$2^{O(n)}$



$+2^{O(n)}$ Pre-processing / Non-uniform algorithm

Extensions

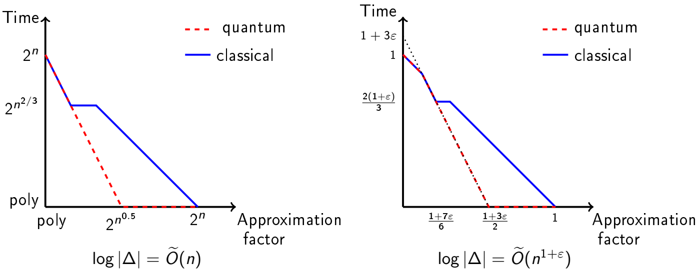
We can extend the algorithm to

- Non-principal ideals

Extensions

We can extend the algorithm to

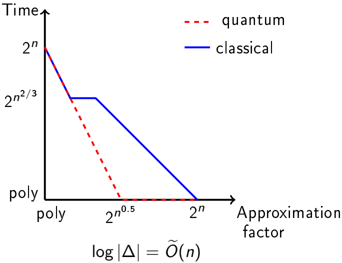
- Non-principal ideals
- All number fields

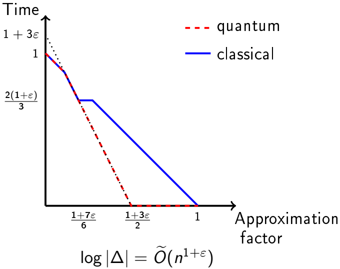


Extensions

We can extend the algorithm to

- Non-principal ideals
- All number fields





Questions?