Approx-SVP in Ideal lattices with Pre-Processing

Alice Pellet--Mary and Damien Stehlé

LIP, ENS de Lyon

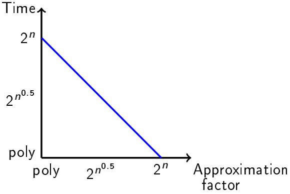
Caen 2018, June 20

ENS DE LYON

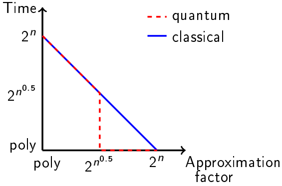
European Research Council
Established by the European Commission

What is this talk about

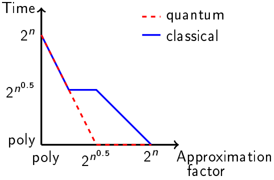
Time/Approximation factor trade-off for SVP in ideal lattices:



BKZ algorithm

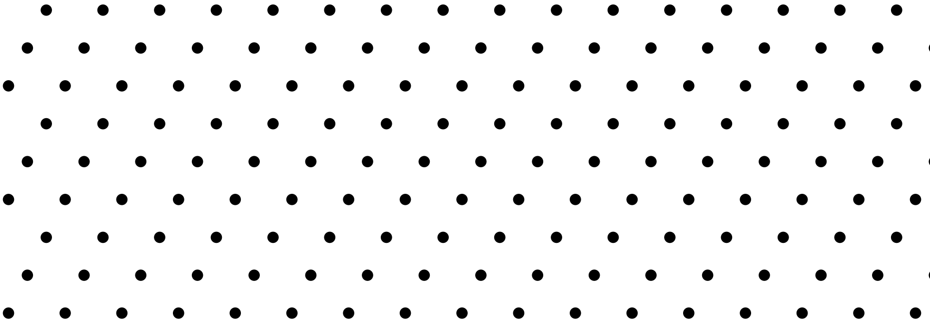


[CDPR16,CDW17]



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

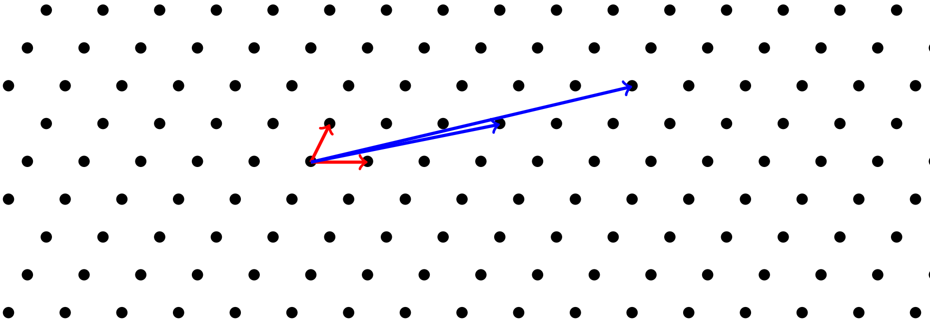
Lattices



Lattice

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

Lattices

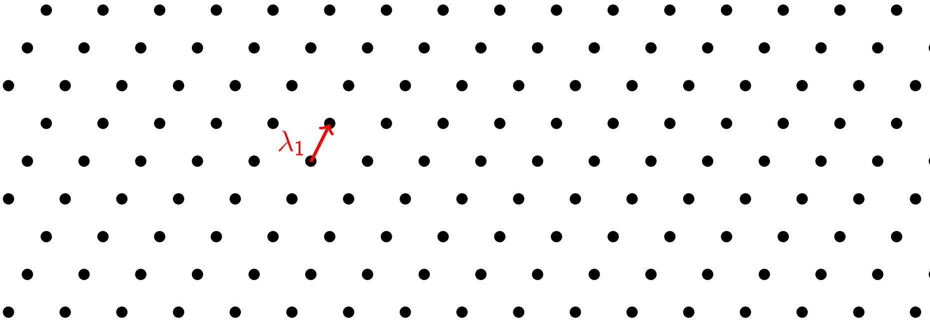


Lattice

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 $\begin{pmatrix} 17 & 11 \\ 4 & 2 \end{pmatrix}$ are two bases of the above lattice.

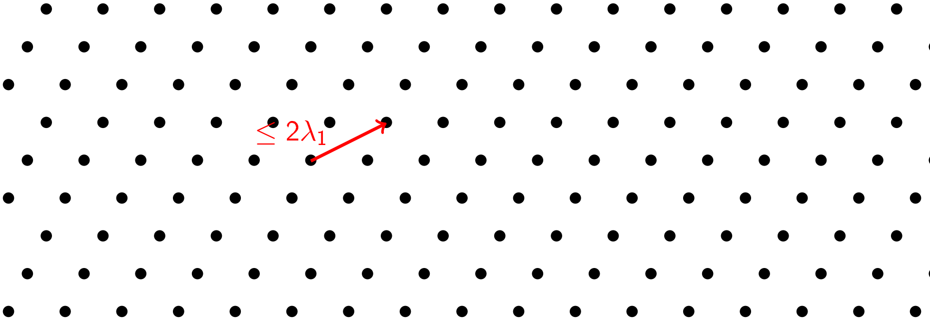
Lattices



Shortest Vector Problem (SVP)

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

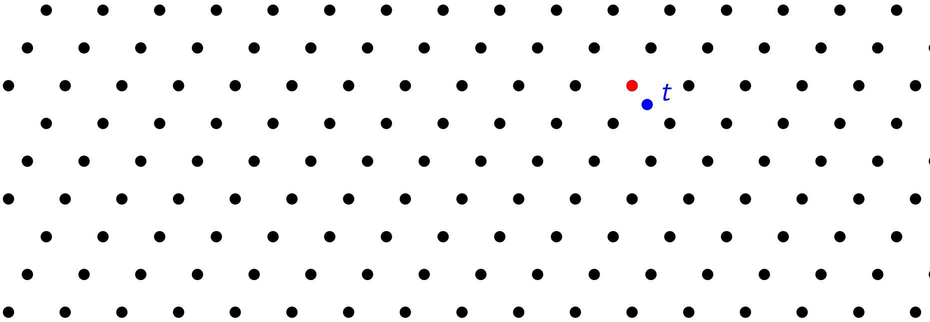
Lattices



Approximate Shortest Vector Problem (approx-SVP)

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

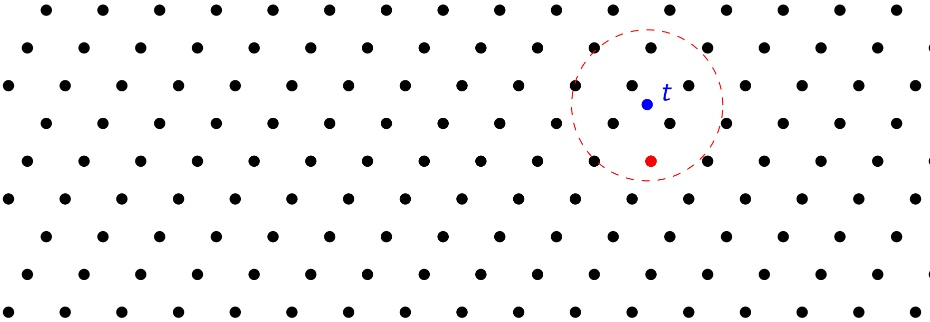
Lattices



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

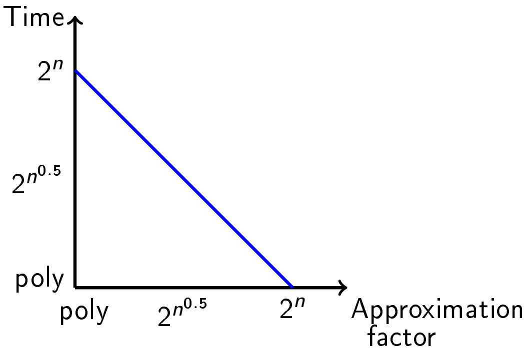
SVP and CVP in general lattices are conjectured to be hard to solve both quantumly and classically $\Rightarrow$ used in cryptography

Complexity of SVP/CVP

Applications

SVP and CVP in general lattices are conjectured to be hard to solve both quantumly and classically $\Rightarrow$ used in cryptography

Best Time/Approximation trade-off for general lattices: BKZ algorithm



Structured lattices

Improve efficiency of lattice-based crypto using structured lattices.
$\Rightarrow$ E.g. ideal lattices

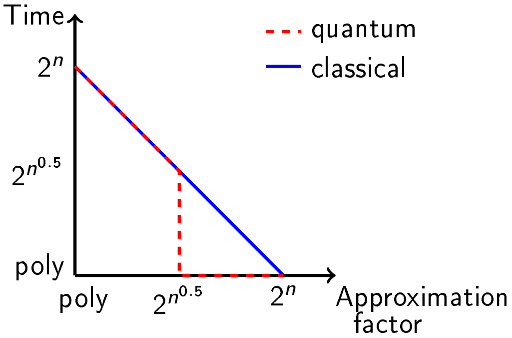
Structured lattices

Improve efficiency of lattice-based crypto using structured lattices.
$\Rightarrow$ E.g. ideal lattices

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

SVP in ideal lattices

[CDPR16,CDW17]: Better than BKZ in the quantum setting

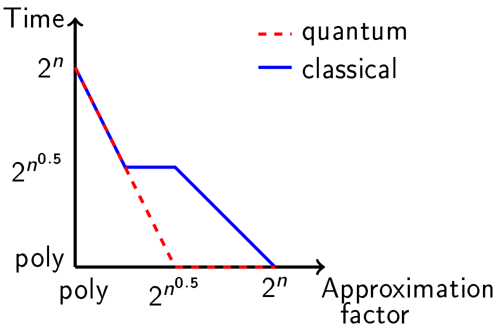


- Heuristic

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

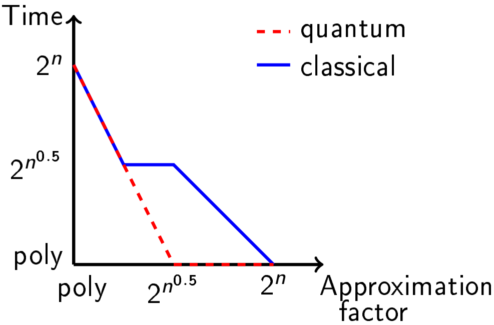
[CDW17] R. Cramer, L. Ducas, B. Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP, Eurocrypt.

This work



- Heuristic
- Pre-processing $2^{O(n)}$, independent of the choice of the ideal (non-uniform algorithm).

This work



- Heuristic
- Pre-processing $2^{O(n)}$, independent of the choice of the ideal (non-uniform algorithm).

Disclaimer: In this talk, only *principal* ideal lattices

Outline of the talk

First definitions

Notation

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

First definitions

Notation

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

- Units: $R^\times = \{a \in R \mid \exists b \in R, ab = 1\}$

First definitions

Notation

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

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

First definitions

Notation

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

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

Minkowski's embedding

- $\zeta \in \mathbb{C}$ primitive $2n$-th root of unity ($\zeta^{2n} = 1$)
- For $r \in R$, define

$$\sigma(r) = (r(\zeta), r(\zeta^3), \cdots, r(\zeta^{n-1})) \in \mathbb{C}^{n/2} \cong \mathbb{R}^n$$

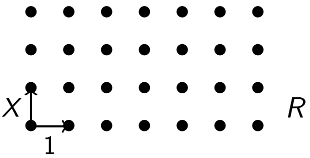
Geometric and algebraic structures

Notation

$$\sigma(r) = (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}}) \in \mathbb{C}^{n/2}$$

Geometric structure:

- Euclidean norm: $\|r\| = \sqrt{\sum_{i=1}^{n/2} |\widetilde{r_i}|^2}$
- $R \subset \mathbb{C}^{n/2} \cong \mathbb{R}^n$ is a lattice



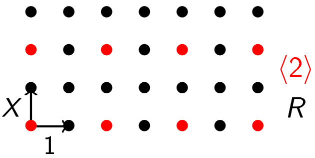
Geometric and algebraic structures

Notation

$$\sigma(r) = (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}}) \in \mathbb{C}^{n/2}$$

Geometric structure:

- Euclidean norm: $\|r\| = \sqrt{\sum_{i=1}^{n/2} |\widetilde{r_i}|^2}$
- $R \subset \mathbb{C}^{n/2} \cong \mathbb{R}^n$ is a lattice
- $\langle g \rangle$ is a sub-lattice of R.



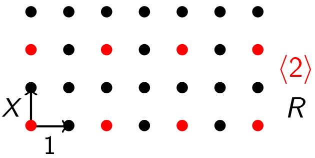
Geometric and algebraic structures

Notation

$$\sigma(r) = (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}}) \in \mathbb{C}^{n/2}$$

Geometric structure:

- Euclidean norm: $\|r\| = \sqrt{\sum_{i=1}^{n/2} |\widetilde{r_i}|^2}$
- $R \subset \mathbb{C}^{n/2} \cong \mathbb{R}^n$ is a lattice
- $\langle g \rangle$ is a sub-lattice of R.



Algebraic structure:

- Algebraic norm: $\mathcal{N}(r) = \prod_{i=1}^{n/2} |\widetilde{r_i}|^2 \in \mathbb{R}$.

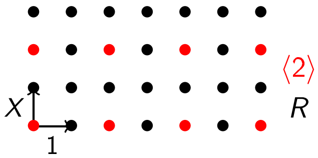
Geometric and algebraic structures

Notation

$$\sigma(r) = (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}}) \in \mathbb{C}^{n/2}$$

Geometric structure:

- Euclidean norm: $\|r\| = \sqrt{\sum_{i=1}^{n/2} |\widetilde{r_i}|^2}$
- $R \subset \mathbb{C}^{n/2} \cong \mathbb{R}^n$ is a lattice
- $\langle g \rangle$ is a sub-lattice of R.



Algebraic structure:

- Algebraic norm: $\mathcal{N}(r) = \prod_{i=1}^{n/2} |\widetilde{r_i}|^2 \in \mathbb{R}$.
  - $\mathcal{N}(ab) = \mathcal{N}(a) \cdot \mathcal{N}(b)$ for all $a, b \in R$,
  - $\mathcal{N}(a) \geq 1$ and $\mathcal{N}(a) \in \mathbb{Z}$ for all $a \in R \setminus \{0\}$,
  - $\mathcal{N}(u) = 1 \iff u \in R^{\times}$.

Relations between algebraic/geometric structures

Reminder: $\sigma(r) = (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}})$

- $\|r\| = \sqrt{\sum_i |\widetilde{r_i}|^2}$
- $\mathcal{N}(r) = \prod_i |\widetilde{r_i}|^2$

Relations between algebraic/geometric structures

Reminder: $\sigma(r) = (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}})$

- $\|r\| = \sqrt{\sum_i |\widetilde{r_i}|^2}$
- $\mathcal{N}(r) = \prod_i |\widetilde{r_i}|^2$

- Euclidean/algebraic norm:
  - $\|r\|$ small $\Rightarrow \mathcal{N}(r)$ relatively small.
  - $\mathcal{N}(r)$ small $\not\Rightarrow \|r\|$ relatively small (e.g. $(2^{-50}, 2^{50})$).

Relations between algebraic/geometric structures

Reminder: $\sigma(r) = (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}})$

- $\|r\| = \sqrt{\sum_i |\widetilde{r_i}|^2}$
- $\mathcal{N}(r) = \prod_i |\widetilde{r_i}|^2$

- Euclidean/algebraic norm:
  - $\|r\|$ small $\Rightarrow \mathcal{N}(r)$ relatively small.
  - $\mathcal{N}(r)$ small $\not\Rightarrow \|r\|$ relatively small (e.g. $(2^{-50}, 2^{50})$).

- $\lambda_1(\langle g \rangle) = \mathrm{poly}(n) \cdot \mathcal{N}(g)^{1/n}$

Objective of this talk

Objective

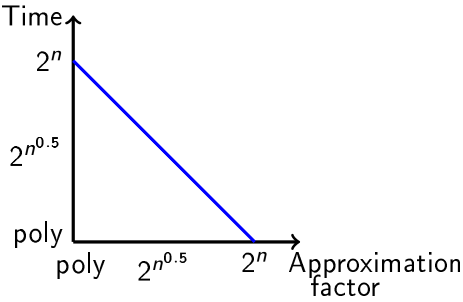
Given a basis of a principal ideal $\langle g \rangle$ and $\alpha \in (0, 1]$,
Find $r \in \langle g \rangle$ such that $\|r\| \leq 2^{\widetilde{O}(n^{\alpha})} \cdot \lambda_1 = 2^{\widetilde{O}(n^{\alpha})} \cdot \mathcal{N}(g)^{1/n}$.

Objective of this talk

Objective

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

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



Outline of the talk

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

Overview of the CDPR algorithm (on an idea of [CGS14])

Important points

- Large algebraic norm $\Rightarrow$ large Euclidean norm.
- In $\langle g \rangle$, the elements with the smallest algebraic norm are the generators.

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

[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.

Overview of the CDPR algorithm (on an idea of [CGS14])

Important points

- Large algebraic norm $\Rightarrow$ large Euclidean norm.
- In $\langle g \rangle$, the elements with the smallest algebraic norm are the generators.

The CDPR algorithm: find a generator with a smallest Euclidean norm

---

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

[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.

Overview of the CDPR algorithm (on an idea of [CGS14])

Important points

- Large algebraic norm $\Rightarrow$ large Euclidean norm.
- In $\langle g \rangle$, the elements with the smallest algebraic norm are the generators.

The CDPR algorithm: find a generator with a smallest Euclidean norm

- Find a generator g_1 of $\langle g \rangle$
  - [BS16]: quantum time $\mathrm{poly}(n)$
  - [BEFGK17]: classical time $2^{\widetilde{O}(\sqrt{n})}$
- Find $u \in R^\times$ which minimizes $\|ug_1\|$.

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

[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.

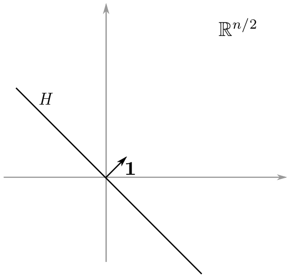
[BEFGK17]: J.F. Biasse, T. Espitau, P.A. Fouque, A. Gélin, P. Kirchner. Computing generator in cyclotomic integer rings, Eurocrypt.

The Log unit lattice

Definitions

$$\begin{aligned} \mathrm{Log} : \sigma(R) &\to \mathbb{R}^{n/2} \\ (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}}) &\mapsto (\log|\widetilde{r_1}|, \cdots, \log|\widetilde{r_{n/2}}|) \end{aligned}$$

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



The Log unit lattice

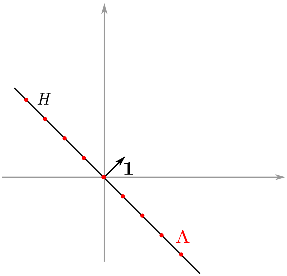
Definitions

$$\begin{aligned} \mathrm{Log} : \sigma(R) &\to \mathbb{R}^{n/2} \\ (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}}) &\mapsto (\log|\widetilde{r_1}|, \cdots, \log|\widetilde{r_{n/2}}|) \end{aligned}$$

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

Theorem (Dirichlet)

$\Lambda := \mathrm{Log}(R^{\times})$ is a lattice included in H.



The Log unit lattice

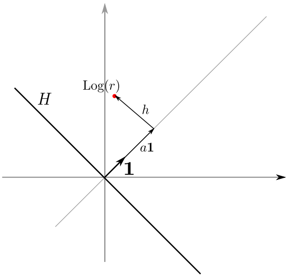
Definitions

$$\begin{aligned} \mathrm{Log} : \sigma(R) &\to \mathbb{R}^{n/2} \\ (\widetilde{r_1}, \cdots, \widetilde{r_{n/2}}) &\mapsto (\log|\widetilde{r_1}|, \cdots, \log|\widetilde{r_{n/2}}|) \end{aligned}$$

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

Theorem (Dirichlet)

$\Lambda := \mathrm{Log}(R^{\times})$ is a lattice included in H.



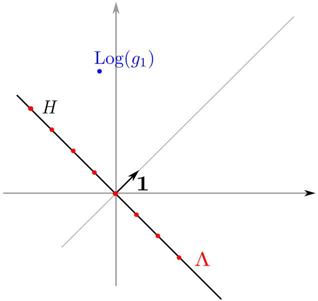
Write $\boxed{\mathrm{Log}(r) = h + a\mathbf{1}}$, with $h \in H$

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \dfrac{\log|\mathcal{N}(r)|}{n}$

CDPR (upper bound)

Reminder ($\mathrm{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$



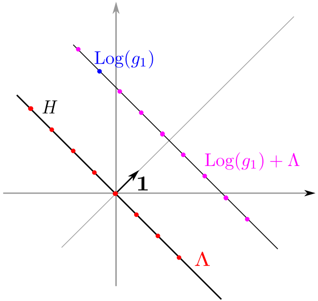
The CDPR Algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]

CDPR (upper bound)

Reminder ($\text{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$



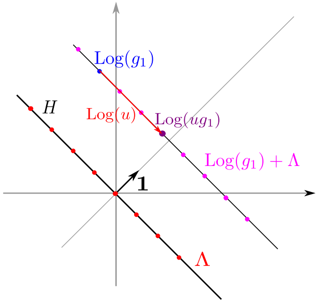
The CDPR Algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]

CDPR (upper bound)

Reminder ($\text{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$



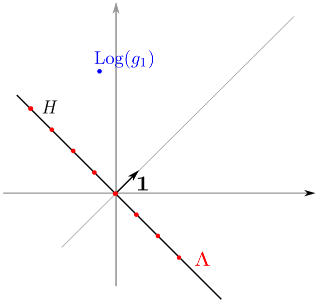
The CDPR Algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]

CDPR (upper bound)

Reminder ($\text{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$



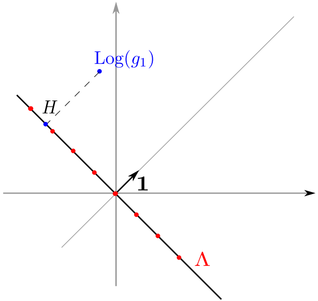
The CDPR Algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]

CDPR (upper bound)

Reminder ($\mathrm{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$



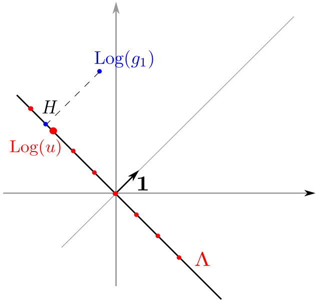
The CDPR Algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]

CDPR (upper bound)

Reminder ($\mathrm{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log|\mathcal{N}(r)|}{n}$



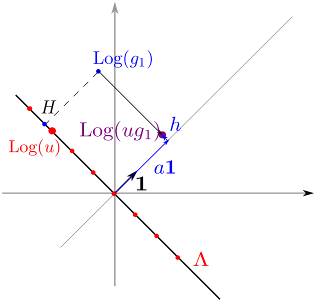
The CDPR Algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]
- Solve CVP in Λ.

CDPR (upper bound)

Reminder ($\text{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log|\mathcal{N}(r)|}{n}$



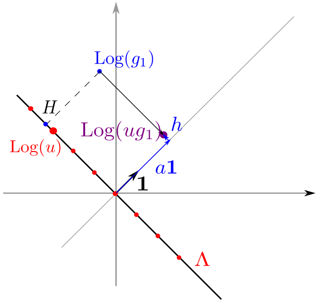
The CDPR Algorithm:

- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]
- Solve CVP in Λ.

CDPR (upper bound)

Reminder ($\mathrm{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$



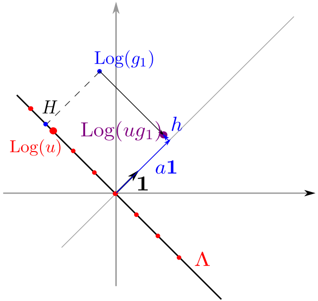
The CDPR Algorithm:

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

CDPR (upper bound)

Reminder ($\text{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$



The CDPR Algorithm:

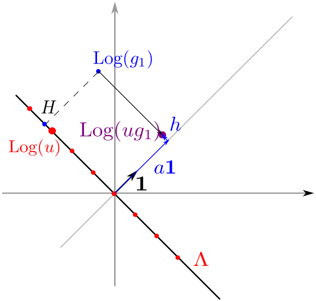
- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]
- Solve CVP in Λ.
  - Good basis of Λ
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$

$$\|ug_1\| \leq \mathcal{N}(ug_1)^{1/n} \cdot 2^{\widetilde{O}(\sqrt{n})}$$

CDPR (upper bound)

Reminder ($\mathrm{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$
- $\lambda_1 = \mathrm{poly}(n) \cdot \mathcal{N}(g)^{1/n}$



The CDPR Algorithm:

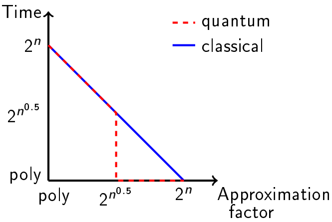
- Find a generator g_1 of $\langle g \rangle$.
  - quantum poly time [BS16]
- Solve CVP in Λ.
  - Good basis of Λ
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$

$$\|ug_1\| \leq \mathcal{N}(ug_1)^{1/n} \cdot 2^{\widetilde{O}(\sqrt{n})}$$
$$\leq 2^{\widetilde{O}(\sqrt{n})} \cdot \lambda_1$$

CDPR (upper bound)

Reminder ($\mathrm{Log}(r) = h + a\mathbf{1}$)

- $\|r\| \leq \sqrt{n} \cdot 2^a \cdot 2^{\|h\|}$
- $a = \frac{\log |\mathcal{N}(r)|}{n}$
- $\lambda_1 = \mathrm{poly}(n) \cdot \mathcal{N}(g)^{1/n}$



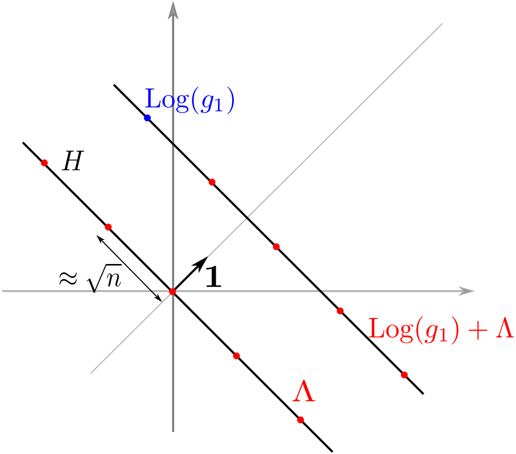
The CDPR Algorithm:

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

$$\|ug_1\| \leq \mathcal{N}(ug_1)^{1/n} \cdot 2^{\widetilde{O}(\sqrt{n})}$$
$$\leq 2^{\widetilde{O}(\sqrt{n})} \cdot \lambda_1$$

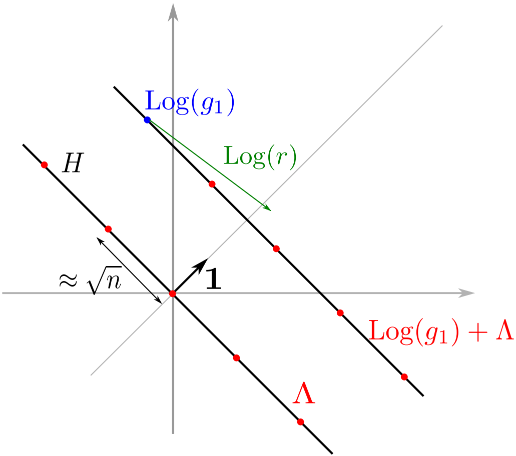
Outline of the talk

Idea

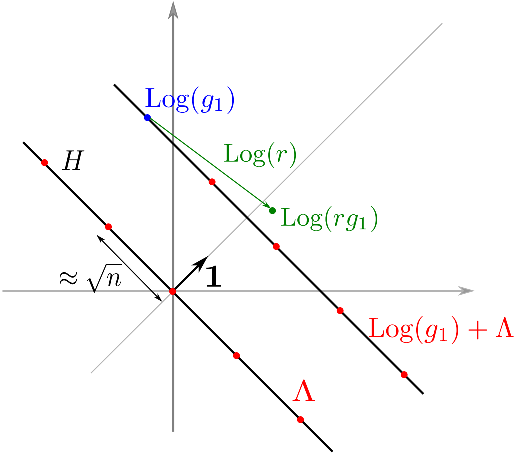


$\mathrm{Log}(g_1)$
H
$\approx \sqrt{n}$
$\mathbf{1}$
$\mathrm{Log}(g_1) + \Lambda$
Λ

Idea

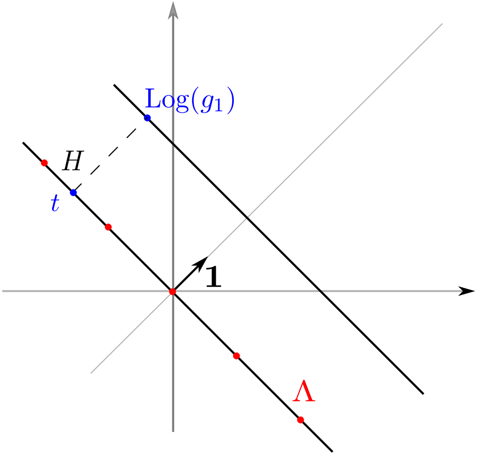


Idea

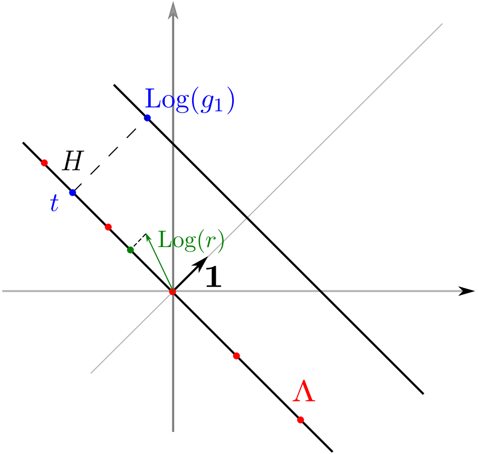


$\mathrm{Log}(g_1)$
$\mathrm{Log}(r)$
H
$\mathrm{Log}(rg_1)$
$\approx \sqrt{n}$
$\mathbf{1}$
$\mathrm{Log}(g_1) + \Lambda$
Λ

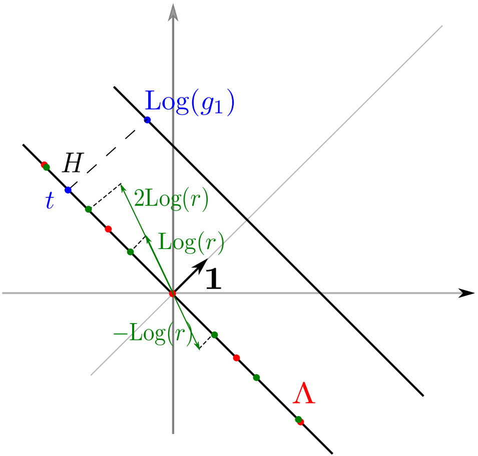
Idea



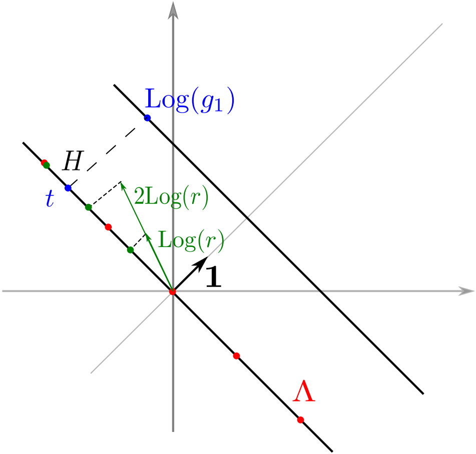
Idea



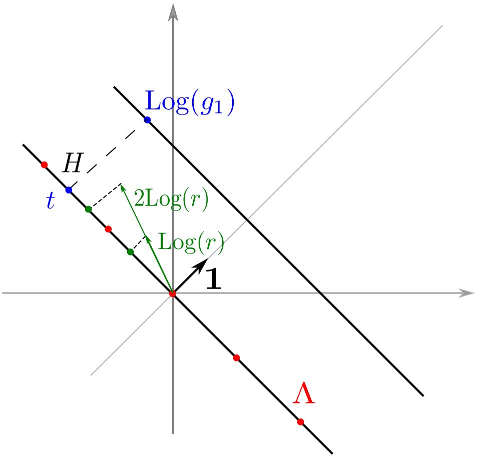
Idea



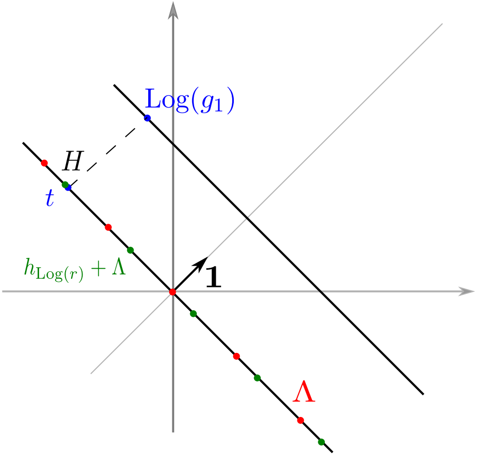
Idea



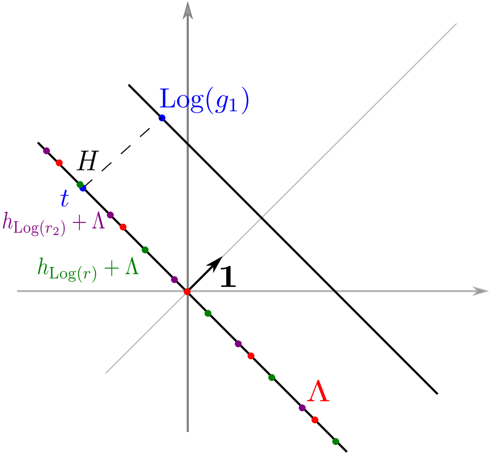
Idea



Idea



Idea

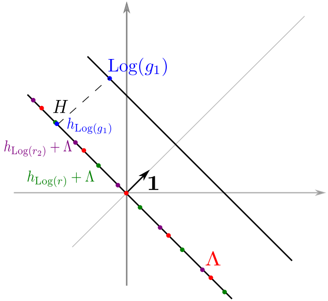

$\mathrm{Log}(g_1)$
H
t
$h_{\mathrm{Log}(r_2)} + \Lambda$
$h_{\mathrm{Log}(r)} + \Lambda$
$\mathbf{1}$
Λ

Formalisation

Difficulties

- We cannot subtract $\mathrm{Log}(r_i)$
- We cannot add too many $\mathrm{Log}(r_i)$'s

$\Rightarrow$ This is not a lattice



Formalisation

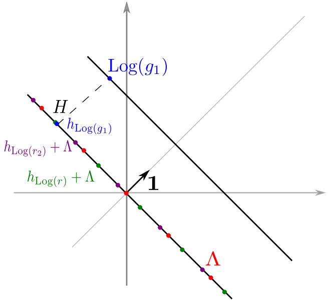
Difficulties

- We cannot subtract $\mathrm{Log}(r_i)$
- We cannot add too many $\mathrm{Log}(r_i)$'s

$\Rightarrow$ This is not a lattice

We consider the lattice

Λ	$h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n}$
0	$\begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix}$

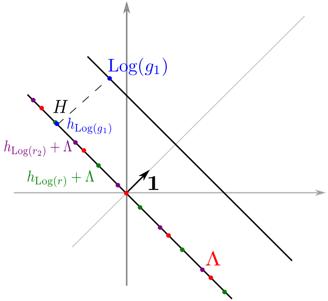


Formalisation

Difficulties

- We cannot subtract $\mathrm{Log}(r_i)$
- We cannot add too many $\mathrm{Log}(r_i)$'s

$\Rightarrow$ This is not a lattice



We consider the lattice and CVP target

Λ	$h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n}$
0	$\begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix}$

$-h_{\mathrm{Log}\, g_1}$
small > 0

Summary

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms

Compute g_1 a generator of $\langle g \rangle$

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms

Compute g_1 a generator of $\langle g \rangle$

Construct $L := \left[\begin{array}{c|c} \Lambda & h_{\mathrm{Log}\, r_1}, \ldots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \end{array}\right]$ and $t := \left[\begin{array}{c} -h_{\mathrm{Log}\, g_1} \\ \hline c > 0 \end{array}\right]$

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms

Compute g_1 a generator of $\langle g \rangle$

Construct $L := \left[\begin{array}{c|c} \Lambda & h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \end{array}\right]$ and $t := \left[\begin{array}{c} -h_{\mathrm{Log}\, g_1} \\ \hline c > 0 \end{array}\right]$

Solve CVP in L with target t (for some $\alpha \in [0, 1]$)
$\Rightarrow$ get a vector $s \in L$ such that $\|s - t\| \leq \widetilde{O}(n^\alpha)$

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms

Compute g_1 a generator of $\langle g \rangle$

Construct $L := \begin{array}{|c|c|} \hline \Lambda & h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \\ \hline \end{array}$ and $t := \begin{bmatrix} -h_{\mathrm{Log}\, g_1} \\ c > 0 \end{bmatrix}$

Solve CVP in L with target t (for some $\alpha \in [0, 1]$)
$\Rightarrow$ get a vector $s \in L$ such that $\|s - t\| \leq \widetilde{O}(n^\alpha)$

Write $s = \begin{bmatrix} h_{\mathrm{Log}\, r} \\ \star \end{bmatrix}$ for some $r \in R$

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms

Compute g_1 a generator of $\langle g \rangle$

Construct $L := \left[\begin{array}{c|c} \Lambda & h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \end{array}\right]$ and $t := \left[\begin{array}{c} -h_{\mathrm{Log}\, g_1} \\ \hline c > 0 \end{array}\right]$

Solve CVP in L with target t (for some $\alpha \in [0, 1]$)
$\Rightarrow$ get a vector $s \in L$ such that $\|s - t\| \leq \widetilde{O}(n^{\alpha})$

Write $s = \left[\begin{array}{c} h_{\mathrm{Log}\, r} \\ \hline \star \end{array}\right]$ for some $r \in R$

$$\|rg_1\| \leq 2^{\widetilde{O}(n^{\alpha})} \cdot \lambda_1$$

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms $\mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Compute g_1 a generator of $\langle g \rangle$ $\mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Construct $L := \left[\begin{array}{c|cccc} \Lambda & h_{\mathrm{Log}\, r_1}, & \dots, & h_{\mathrm{Log}\, r_n} & \\ \hline & 1 & & & \\ 0 & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{array}\right]$ and $t := \left[\begin{array}{c} -h_{\mathrm{Log}\, g_1} \\ \hline c > 0 \end{array}\right]$

Solve CVP in L with target t (for some $\alpha \in [0,1]$)
$\Rightarrow$ get a vector $s \in L$ such that $\|s - t\| \leq \widetilde{O}(n^\alpha)$

Write $s = \left[\begin{array}{c} h_{\mathrm{Log}\, r} \\ \hline \star \end{array}\right]$ for some $r \in R$

$$\|rg_1\| \leq 2^{\widetilde{O}(n^\alpha)} \cdot \lambda_1$$

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms $\quad \mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Compute g_1 a generator of $\langle g \rangle$ $\quad \mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Construct $L := \left[\begin{array}{c|c} \Lambda & h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & \\ & 1 & \\ & & \ddots \\ & & & 1 \end{matrix} \end{array}\right]$ and $t := \left[\begin{array}{c} -h_{\mathrm{Log}\, g_1} \\ \hline c > 0 \end{array}\right]$ $\quad \mathrm{poly}(n)$

Solve CVP in L with target t (for some $\alpha \in [0,1]$)
$\Rightarrow$ get a vector $s \in L$ such that $\|s - t\| \leq \widetilde{O}(n^{\alpha})$

Write $s = \left[\begin{array}{c} h_{\mathrm{Log}\, r} \\ \hline \star \end{array}\right]$ for some $r \in R$ $\quad \mathrm{poly}(n)$

$$\|rg_1\| \leq 2^{\widetilde{O}(n^{\alpha})} \cdot \lambda_1$$

Summary

Compute $r_1, \cdots, r_n$ of small algebraic norms — $\mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Compute g_1 a generator of $\langle g \rangle$ — $\mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Construct $L := \left[\begin{array}{c|c} \Lambda & h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \end{array}\right]$ and $t := \left[\begin{array}{c} -h_{\mathrm{Log}\, g_1} \\ \hline c > 0 \end{array}\right]$ — $\mathrm{poly}(n)$

Solve CVP in L with target t (for some $\alpha \in [0,1]$) — ?
$\Rightarrow$ get a vector $s \in L$ such that $\|s - t\| \leq \widetilde{O}(n^\alpha)$

Write $s = \left[\begin{array}{c} h_{\mathrm{Log}\, r} \\ \hline \star \end{array}\right]$ for some $r \in R$ — $\mathrm{poly}(n)$

$$\|rg_1\| \leq 2^{\widetilde{O}(n^\alpha)} \cdot \lambda_1$$

How to solve CVP in L?

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

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

How to solve CVP in L?

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

Key observation

$$L := \left[\begin{array}{c|c} \Lambda & h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \end{array}\right]$$

does not depend on $\langle g \rangle$

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

How to solve CVP in L?

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

Key observation

$$L := \left[\begin{array}{c|c} \Lambda & h_{\mathrm{Log}\, r_1}, \dots, h_{\mathrm{Log}\, r_n} \\ \hline 0 & \begin{matrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{matrix} \end{array}\right]$$

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

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

How to solve CVP in L?

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

Key observation

$$L := \left(\begin{array}{c|cccc} \Lambda & h_{\mathrm{Log}\,r_1}, & \dots, & h_{\mathrm{Log}\,r_n} & \\ \hline & 1 & & & \\ 0 & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{array}\right)$$

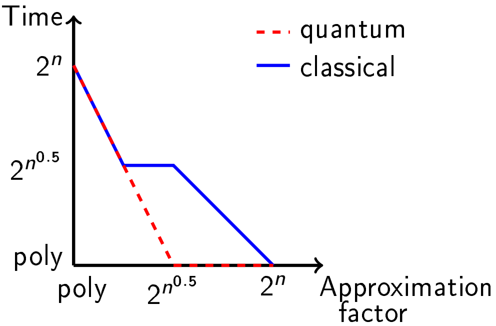
does not depend on $\langle g \rangle$ $\Rightarrow$ Pre-processing on L

[Laa16]:
- 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.

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

- Non principal ideals ✓
- Generalization to other number fields ?
- Removing (or testing) the heuristics ?

Extensions

- Non principal ideals ✓
- Generalization to other number fields ?
- Removing (or testing) the heuristics ?

Questions?