Approx-SVP in Ideal Lattices with Pre-Processing

Alice Pellet-Mary, Guillaume Hanrot and Damien Stehlé

ENS de Lyon

RISC and Prometheus seminar
May 3rd, 2019

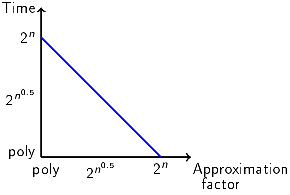
https://eprint.iacr.org/2019/215.pdf

ENS DE LYON

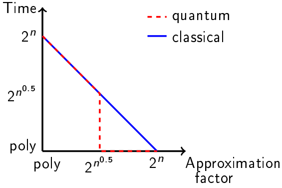
European Research Council
Established by the European Commission

What is this talk about

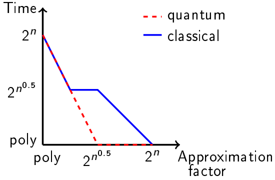
Time/Approximation factor trade-off for SVP in ideal lattices:



BKZ algorithm [Sch87]

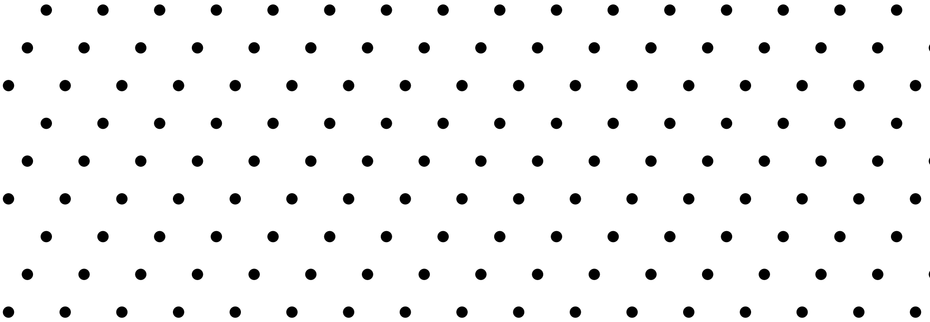


[CDW17]



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

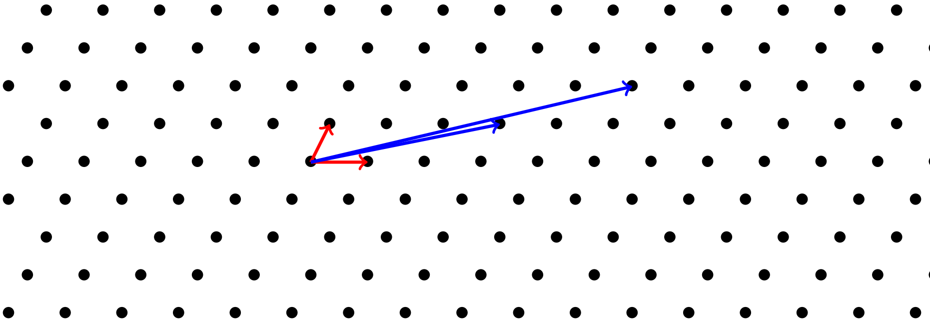
Lattices



Lattice

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

Lattices

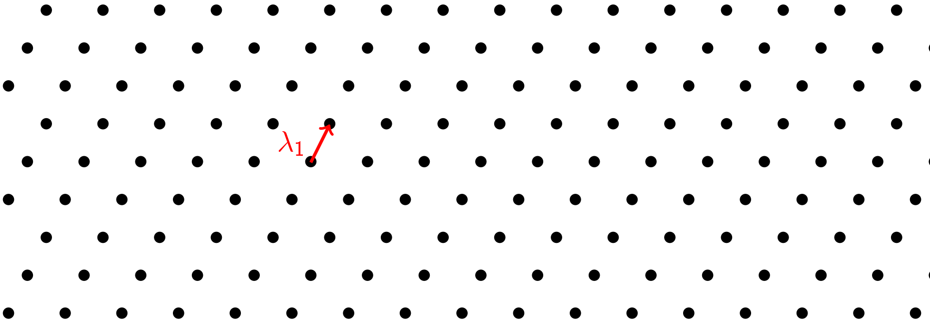


Lattice

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

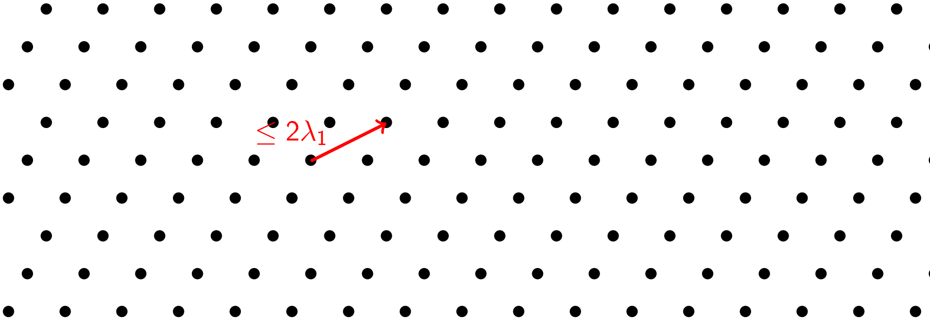
Lattices



Shortest Vector Problem (SVP)

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

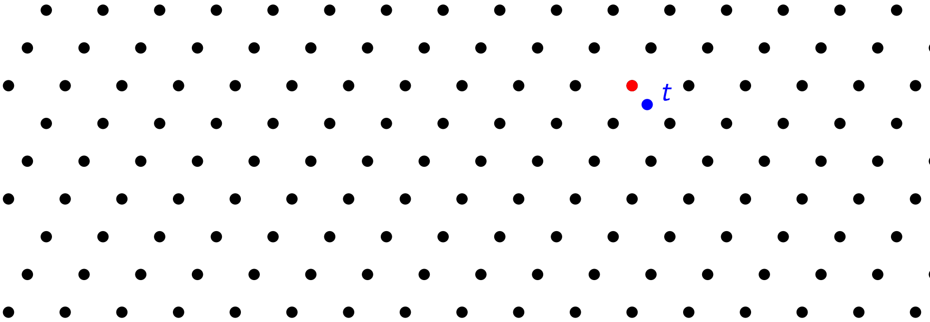
Lattices



Approximate Shortest Vector Problem (approx-SVP)

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

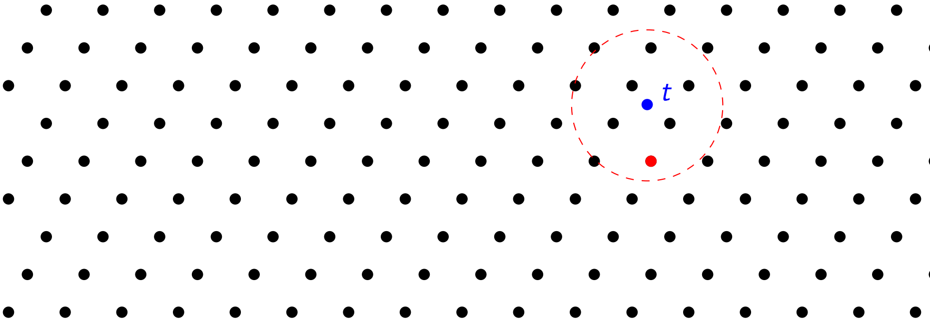
Lattices



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

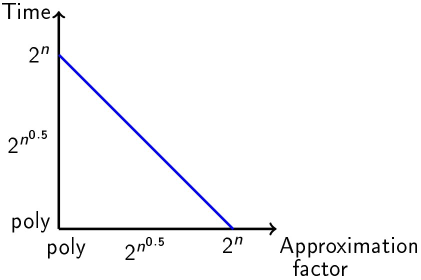
[Sch87] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical computer science.

Complexity of SVP/CVP

Applications

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

Best Time/Approx trade-off for generic 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

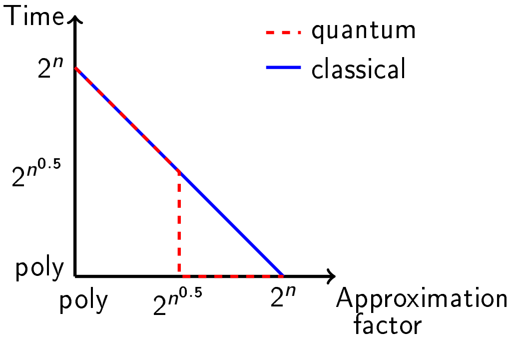
Structured lattices

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

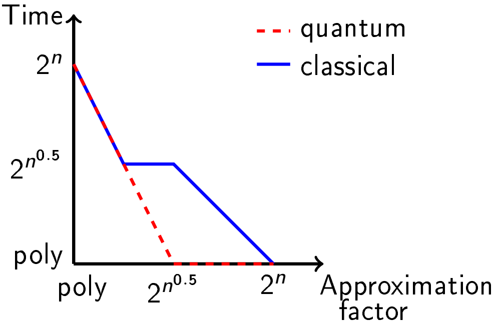
[CDW17]: Better than BKZ in the quantum setting



- 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



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

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]

$$\boxed{\text{schemes}} \longrightarrow \boxed{\text{RLWE}} \longrightarrow \boxed{\text{ideal SVP}}$$

[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]

$$\boxed{\text{schemes}} \longrightarrow \boxed{\text{RLWE}} \underset{?}{\rightleftarrows} \boxed{\text{ideal SVP}}$$

[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]

$$\begin{tikzcd} \boxed{\text{schemes}} \arrow[r] & \boxed{\text{RLWE}} \arrow[r, shift left] \arrow[d, leftrightarrow] & \boxed{\text{ideal SVP}} \arrow[l, shift left, "?"] \\ & \boxed{\text{module SIVP}} & \end{tikzcd}$$

[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
4. Extension: “Euclidean division” over R

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

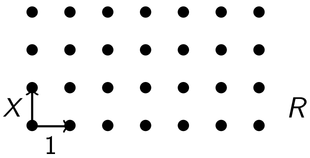
$R = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$ (for simplicity)

- 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?

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

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

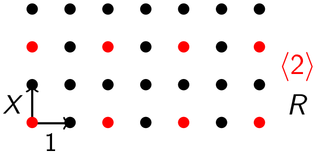


Why is $\langle g \rangle$ a lattice?

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

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

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



Objective of this talk

Objective

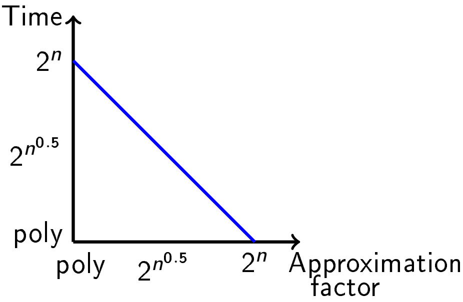
Given a basis of a principal ideal $\langle g \rangle$ and $\alpha \in (0,1]$,
Find $r \in \langle g \rangle$ such that $\|r\| \leq 2^{\widetilde{O}(n^\alpha)} \cdot \lambda_1$.

Objective of this talk

Objective

Given a basis of a principal ideal $\langle g \rangle$ and $\alpha \in (0,1]$,
Find $r \in \langle g \rangle$ 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

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

Idea

Maybe g is a somehow small element of $\langle g \rangle$

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

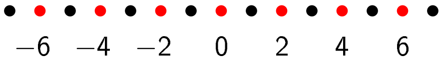
[CGS14]: P. Campbell, M. Groves, and D. Shepherd. Soliloquy: A cautionary tale.

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

Idea

Maybe g is a somehow small element of $\langle g \rangle$

If $\mathbf{n = 1}$: e.g. $\langle 2 \rangle \Rightarrow 2$ and -2 are the smallest elements.



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

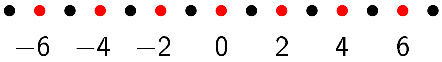
[CGS14]: P. Campbell, M. Groves, and D. Shepherd. Soliloquy: A cautionary tale.

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

Idea

Maybe g is a somehow small element of $\langle g \rangle$

If $\mathbf{n = 1}$: e.g. $\langle 2 \rangle \Rightarrow 2$ and -2 are the smallest elements.



For larger n: one of the generators is somehow small

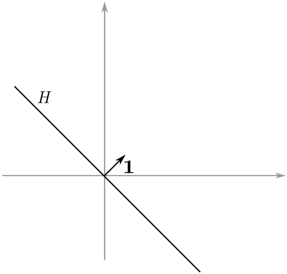
[CDPR16] R. Cramer, L. Ducas, C. Peikert and O. Regev. Recovering Short Generators of Principal Ideals in Cyclotomic Rings, Eurocrypt.

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

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

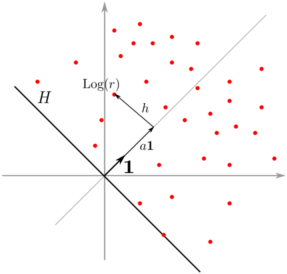
$\text{Log} : R \to \mathbb{R}^n$ (somehow generalising log to R)

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

Properties

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

- $a \geq 0$



The Log space

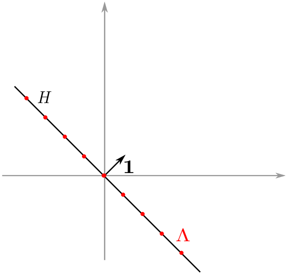
Log : $R \to \mathbb{R}^n$ (somehow generalising log to R)

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

Properties

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

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



The Log space

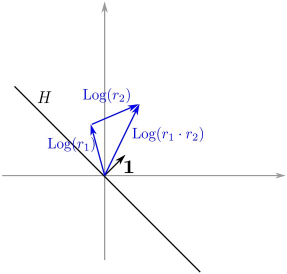
$\mathrm{Log} : R \to \mathbb{R}^n$ (somehow generalising log to R)

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
- $\mathrm{Log}(r_1 \cdot r_2) = \mathrm{Log}(r_1) + \mathrm{Log}(r_2)$



The Log space

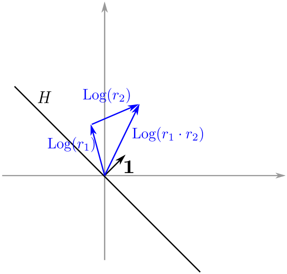
$\text{Log} : R \to \mathbb{R}^n$ (somehow generalising log to R)

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

Properties

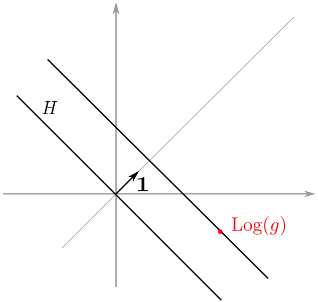
$\text{Log}\, r = h + a\mathbf{1}$, with $h \in H$

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



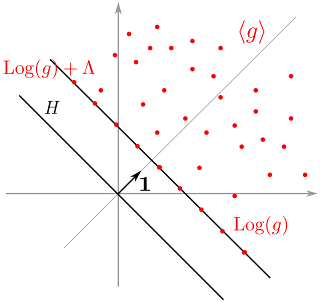
The CDPR algorithm

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



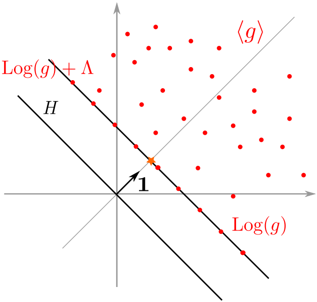
The CDPR algorithm

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



The CDPR algorithm

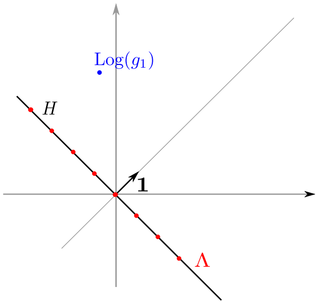
What does $\text{Log}\langle g\rangle$ look like?



The CDPR algorithm

The CDPR Algorithm:

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



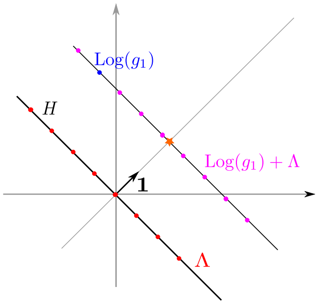
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $2^{\widetilde{O}(\sqrt{n})}$



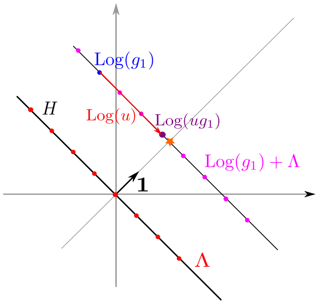
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $2^{\widetilde{O}(\sqrt{n})}$



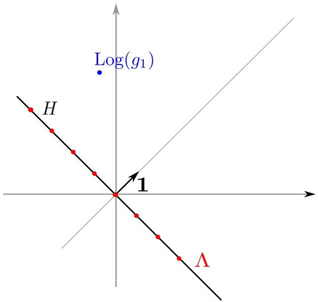
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $2^{\widetilde{O}(\sqrt{n})}$



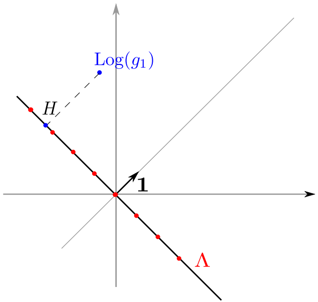
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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



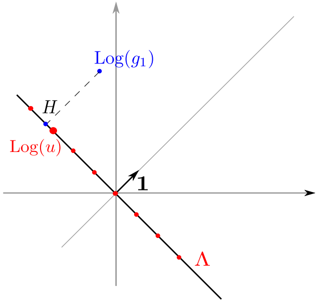
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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



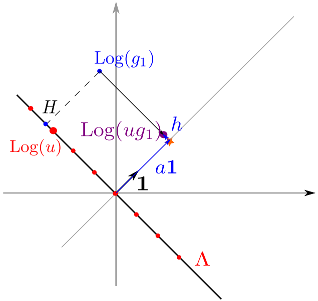
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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



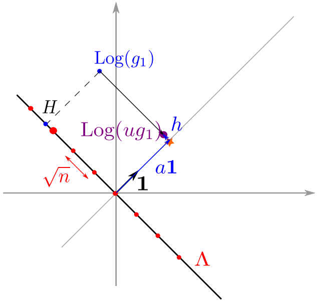
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$



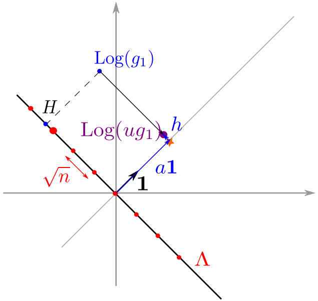
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$



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

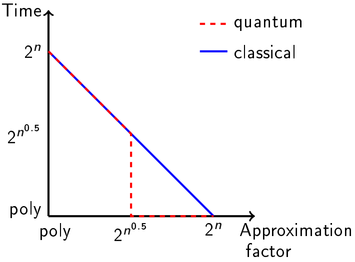
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$



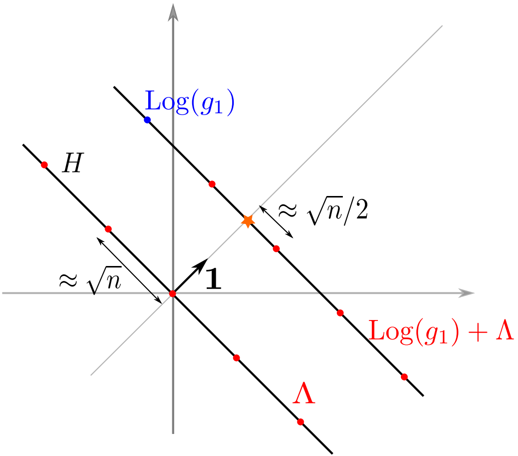
$$\boxed{\|ug_1\| \leq 2^{\widetilde{O}(\sqrt{n})} \cdot \lambda_1}$$

[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.

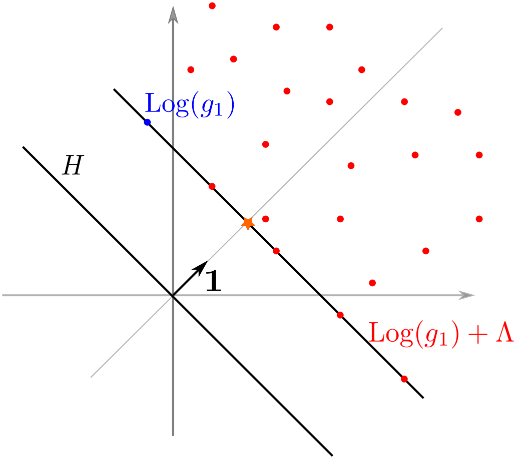
This work

Idea

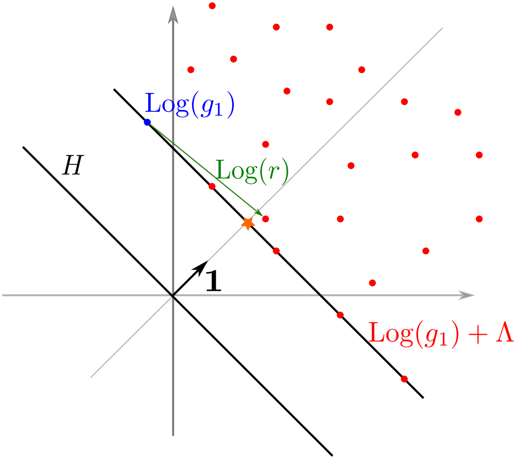


$\mathrm{Log}(g_1)$
H
$\approx \sqrt{n}/2$
$\approx \sqrt{n}$
$\mathbf{1}$
$\mathrm{Log}(g_1) + \Lambda$
Λ

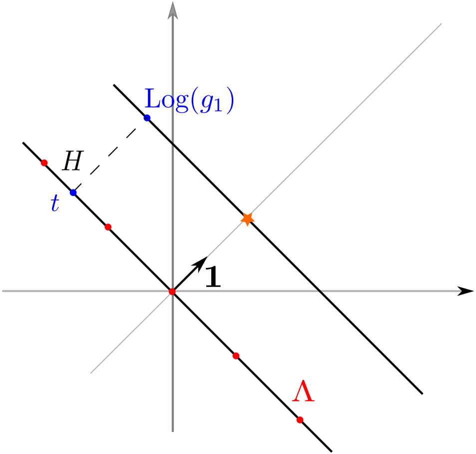
Idea



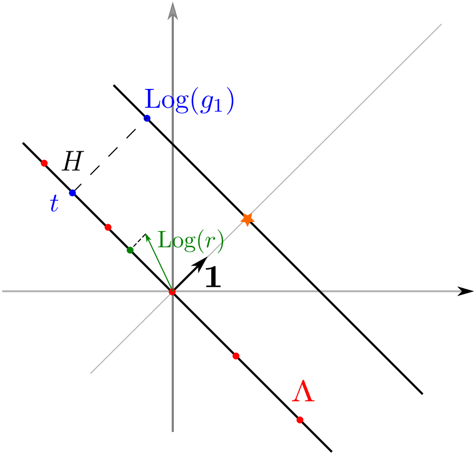
Idea



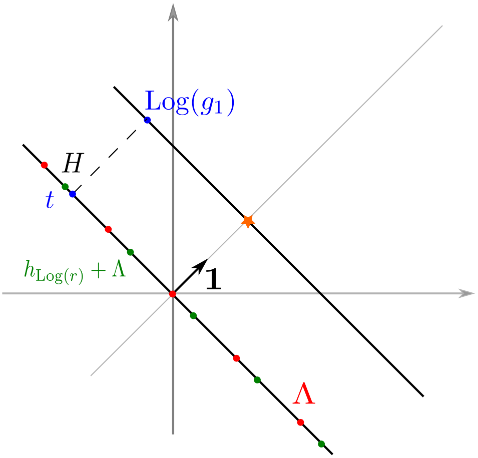
Idea



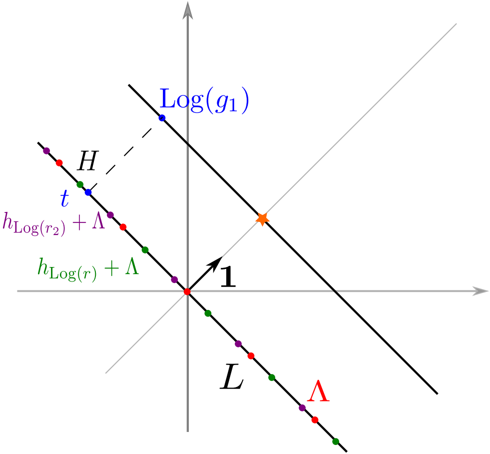
Idea



Idea



Idea



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

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

$L = \Lambda \cup \bigcup_i (h_{\text{Log } r_i} + \Lambda)$ 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

$L = \Lambda \cup \bigcup_i (h_{\mathrm{Log}\, r_i} + \Lambda)$ does not depend on $\langle g \rangle$ $\Rightarrow$ 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

$L = \Lambda \cup \bigcup_i (h_{\mathrm{Log}\, r_i} + \Lambda)$ does not depend on $\langle g \rangle$ $\Rightarrow$ Pre-processing on L

[DLW19,Ste19]:

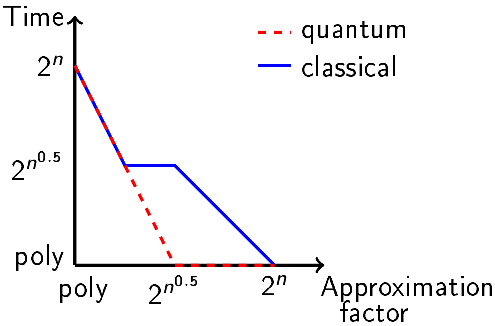
- Find $s \in L$ such that $\|s - t\| = \widetilde{O}(n^{\alpha})$
- Time:
  - $2^{\widetilde{O}(n^{1-2\alpha})}$ (query)
  - $+\ 2^{O(n)}$ (pre-processing)

[DLW19]: E. Doulgerakis, T. Laarhoven, and B. de Weger. Finding closest lattice vectors using approximate Voronoi cells. PQCRYPTO 2019.

[Ste19]: N. Stephens-Davidowitz. A time-distance trade-off for GDD with preprocessing – instantiating the DLW heuristic. arXiv 2019.

Conclusion

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



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

Extensions

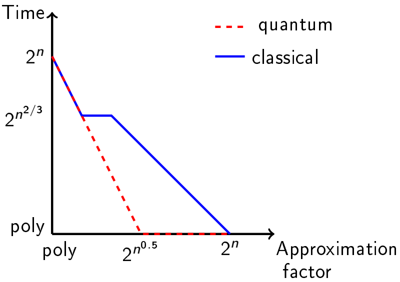
We can extend the algorithm to

- Non-principal ideals

Extensions

We can extend the algorithm to

- Non-principal ideals
- All number fields



Work in progress: "Euclidean division" over R

joint work with
Changmin Lee, Damien Stehlé and Alexandre Wallet

Finding short vectors in module lattices

(Principal) Ideals

Input: $a \in R$
Output: $x \in R$
such that $\|ax\|$ is small

(Free) Modules

Input: $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in R^{2\times 2}$

Output: $(x, y) \in R^2$ such that

$$\left\| \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \right\| = \left\| \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix} \right\|$$

is small

If $R = \mathbb{Z}$: the LLL algorithm (or Gauss/Lagrange in dim 2)

LLL algorithm over $\mathbb{Z}$:

1. $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \xrightarrow{QR} \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}$
2. Reduce $r_{12} \leftarrow r_{12} \bmod r_{11}$
   ($\Rightarrow |r_{12}| \leq |r_{11}|/2$)
3. If $|r_{22}| \leq |r_{11}|/2$
   ($\Rightarrow \sqrt{r_{12}^2 + r_{22}^2} \leq |r_{11}|/\sqrt{2}$)
   - Swap the two columns
   - Go to Step 1

If $R = \mathbb{Z}$: the LLL algorithm (or Gauss/Lagrange in dim 2)

LLL algorithm over $\mathbb{Z}$:

1. $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \xrightarrow{QR} \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}$
2. Reduce $r_{12} \leftarrow r_{12} \bmod r_{11}$
   $(\Rightarrow |r_{12}| \leq |r_{11}|/2)$
3. If $|r_{22}| \leq |r_{11}|/2$
   $(\Rightarrow \sqrt{r_{12}^2 + r_{22}^2} \leq |r_{11}|/\sqrt{2})$
   - Swap the two columns
   - Go to Step 1

Adaptation to R

We need:

- A scalar product $\langle \cdot, \cdot \rangle$
  - and a norm $|\cdot|$
- A division (Step 2)
  - $|r_{12}| < (1 - \varepsilon)|r_{11}|$
  - swap condition $|r_{22}| < \varepsilon |r_{11}|$

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target b/a.

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target b/a.

Over R

CVP in R with target b/a
$\Rightarrow$ output $r \in R$

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target b/a.

Over R

CVP in R with target b/a
$\Rightarrow$ output $r \in R$

Difficulty: Typically
$\|b/a + r\| \approx \sqrt{n} \gg 1$.

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target b/a.

Over R

CVP in R with target b/a
$\Rightarrow$ output $r \in R$

Difficulty: Typically
$\|b/a + r\| \approx \sqrt{n} \gg 1$.

Relax the requirement

Find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(n)$

Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(n)$

Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(n)$

Difficulty: Log works well with $\times$, but not with $+$

Solution: If $\|\operatorname{Log}(u) - \operatorname{Log}(v)\| \leq \varepsilon$
then $\|u - v\| \lesssim \varepsilon \cdot \min(\|u\|, \|v\|)$
(requires to extend Log to take arguments into account)

Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(n)$

Difficulty: Log works well with $\times$, but not with $+$

Solution: If $\|\operatorname{Log}(u) - \operatorname{Log}(v)\| \leq \varepsilon$
then $\|u - v\| \lesssim \varepsilon \cdot \min(\|u\|, \|v\|)$
(requires to extend Log to take arguments into account)

New objective

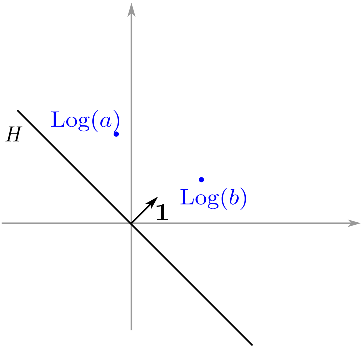
Find $x, y \in R$ such that

- $\|\operatorname{Log}(xa) - \operatorname{Log}(yb)\| \leq \varepsilon$
- $\|\operatorname{Log}(y)\|_\infty \leq O(\log n)$

Idea

Objective: find $x, y \in R$ s.t.

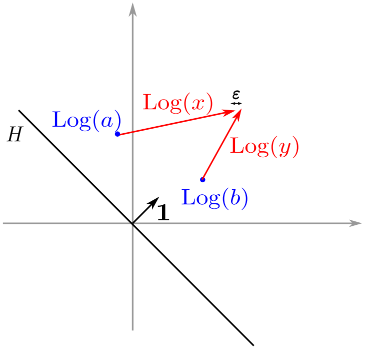
- $\| \mathrm{Log}(xa) - \mathrm{Log}(yb) \| \leq \varepsilon$
- $\| \mathrm{Log}(y) \|_\infty \leq O(\log n)$



Idea

Objective: find $x, y \in R$ s.t.

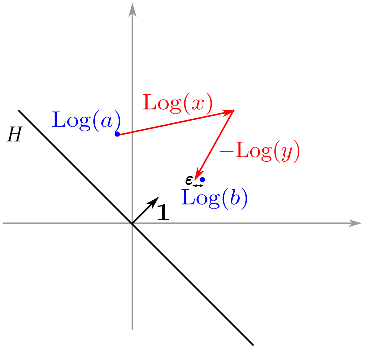
- $\| \mathrm{Log}(xa) - \mathrm{Log}(yb) \| \leq \varepsilon$
- $\| \mathrm{Log}(y) \|_\infty \leq O(\log n)$



Idea

Objective: find $x, y \in R$ s.t.

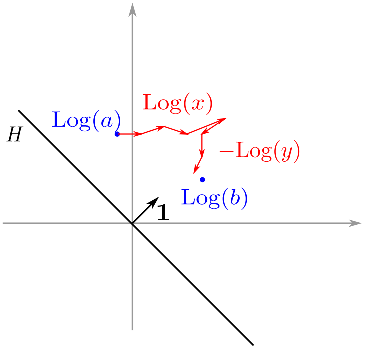
- $\| \mathrm{Log}(xa) - \mathrm{Log}(yb) \| \leq \varepsilon$
- $\| \mathrm{Log}(y) \|_\infty \leq O(\log n)$



Idea

Objective: find $x, y \in R$ s.t.

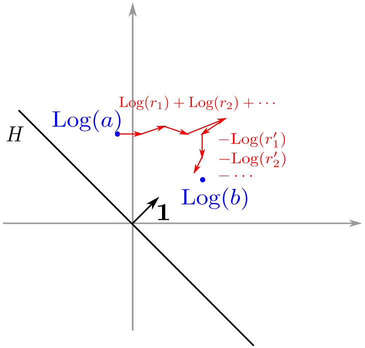
- $\|\operatorname{Log}(xa) - \operatorname{Log}(yb)\| \leq \varepsilon$
- $\|\operatorname{Log}(y)\|_\infty \leq O(\log n)$



Idea

Objective: find $x, y \in R$ s.t.

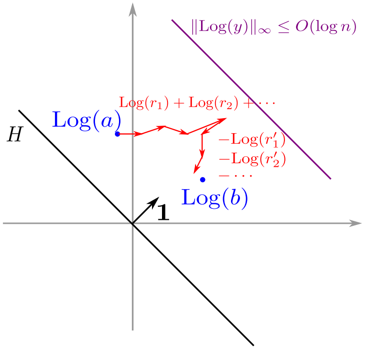
- $\| \mathrm{Log}(xa) - \mathrm{Log}(yb) \| \leq \varepsilon$
- $\| \mathrm{Log}(y) \|_\infty \leq O(\log n)$



Idea

Objective: find $x, y \in R$ s.t.

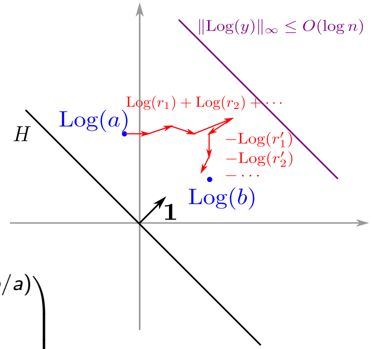
- $\|\mathrm{Log}(xa) - \mathrm{Log}(yb)\| \leq \varepsilon$
- $\|\mathrm{Log}(y)\|_\infty \leq O(\log n)$



Idea

Objective: find $x, y \in R$ s.t.

- $\| \mathrm{Log}(xa) - \mathrm{Log}(yb) \| \leq \varepsilon$
- $\| \mathrm{Log}(y) \|_\infty \leq O(\log n)$



$$L = \begin{pmatrix} \mathrm{Log}\, r_1 & \mathrm{Log}\, r_2 & \cdots & \mathrm{Log}\, r_{n^2} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \quad t = \begin{pmatrix} \mathrm{Log}(b/a) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Conclusion

Difficulty

We need exact CVP in L: approx-CVPP algorithms do not suffice

Conclusion

Difficulty

We need exact CVP in L: approx-CVPP algorithms do not suffice

- We obtain a division in R
  - Using CVP oracle in a lattice L of dim n^2 (depending only on R)
  - Classical time $2^{\tilde{O}(\sqrt{n})}$ / quantum time $\mathrm{poly}(n)$
  - Heuristic

Conclusion

Difficulty

We need exact CVP in L: approx-CVPP algorithms do not suffice

- We obtain a division in R
  - Using CVP oracle in a lattice L of dim n^2 (depending only on R)
  - Classical time $2^{\tilde{O}(\sqrt{n})}$ / quantum time $\mathrm{poly}(n)$
  - Heuristic
- Can be used to adapt LLL algorithm to modules over R

Conclusion

Difficulty

We need exact CVP in L: approx-CVPP algorithms do not suffice

- We obtain a division in R
  - Using CVP oracle in a lattice L of dim n^2 (depending only on R)
  - Classical time $2^{\tilde{O}(\sqrt{n})}$ / quantum time $\mathrm{poly}(n)$
  - Heuristic
- Can be used to adapt LLL algorithm to modules over R

Questions?