# Approx-SVP in Ideal lattices with Pre-Processing

**Alice Pellet-Mary**, Guillaume Hanrot and Damien Stehlé

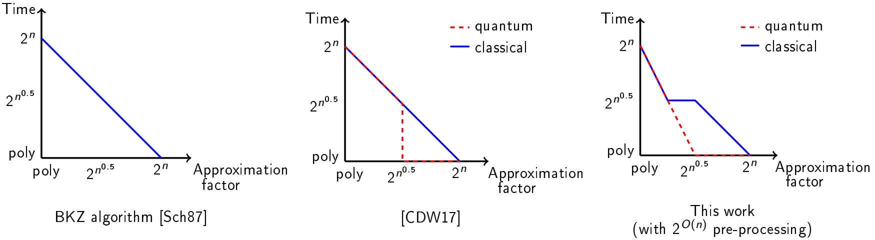
LIP, ENS de Lyon

Séminaire de cryptographie de Rennes,
May 24, 2019

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

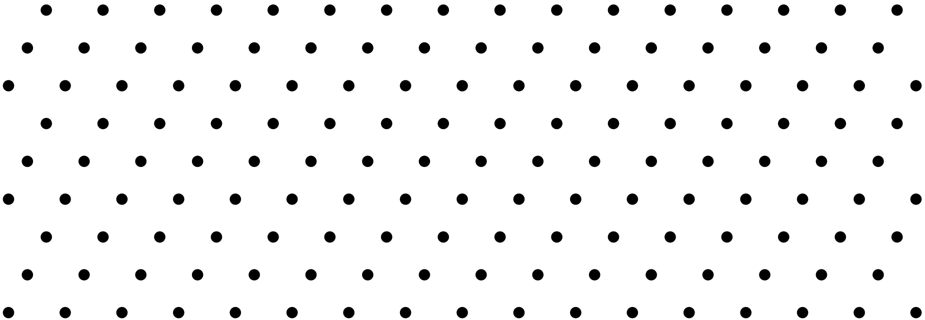
# What is this talk about

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




(Figures are for prime power cyclotomic fields)

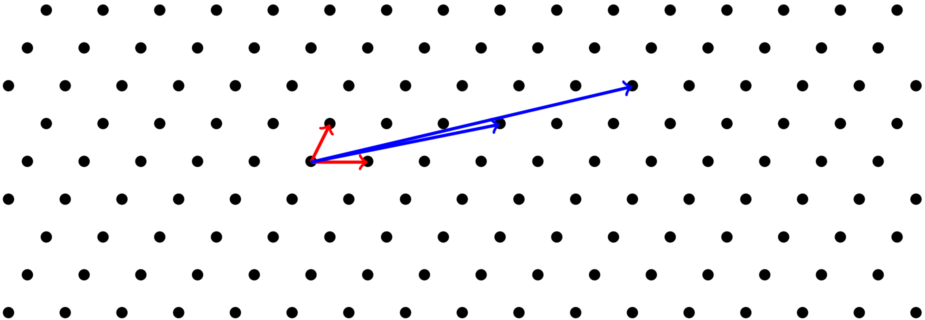
# Lattices



## Lattice

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

# Lattices

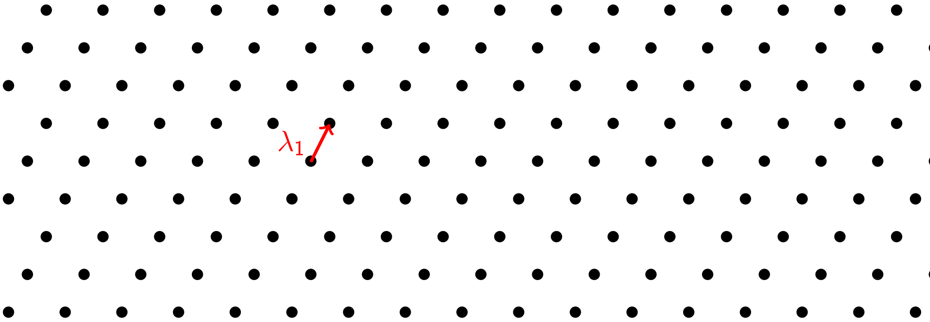


## Lattice

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

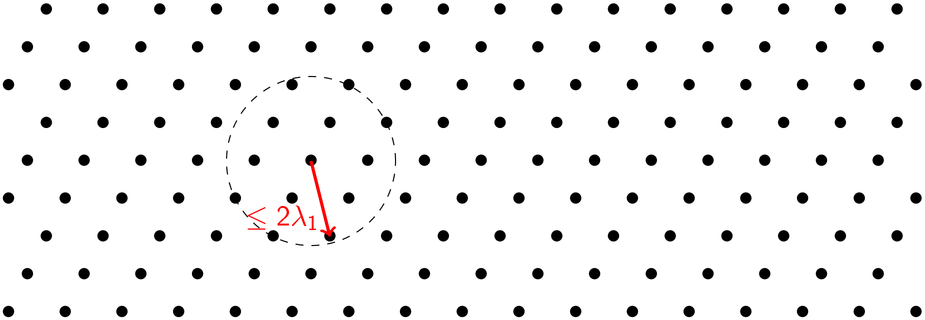
# Lattices




**Shortest Vector Problem (SVP)**

Find a shortest (in Euclidean norm) non-zero vector.
Its Euclidean norm is denoted $\lambda_1$.

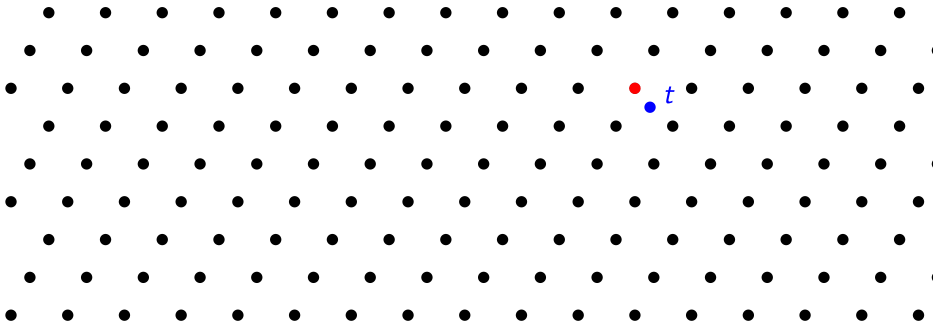
# Lattices




**Approximate Shortest Vector Problem (approx-SVP)**

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

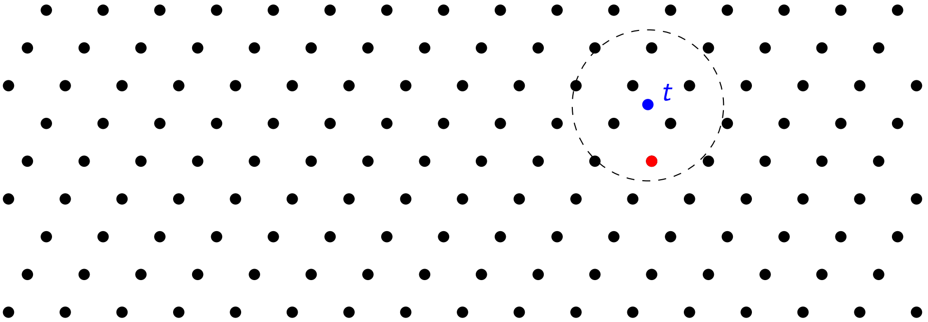
# Lattices




## Closest Vector Problem (CVP)

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

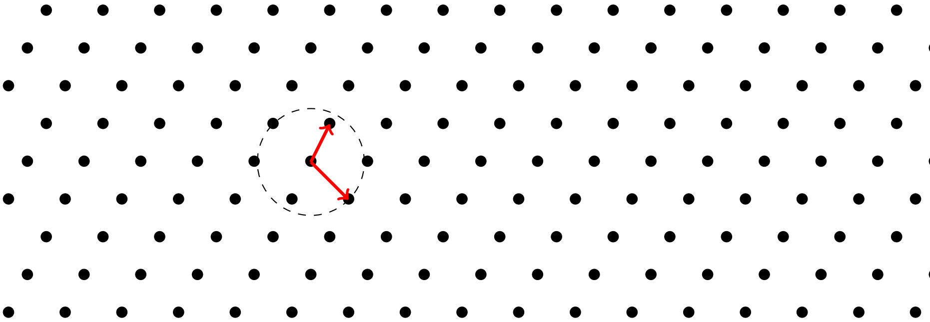
# Lattices




**Approximate Closest Vector Problem (approx-CVP)**

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

# Lattices



## Shortest Independent Vectors Problem (SIVP)

Find $n$ short vectors which are linearly independent.

# Complexity of SVP/CVP

**Applications**

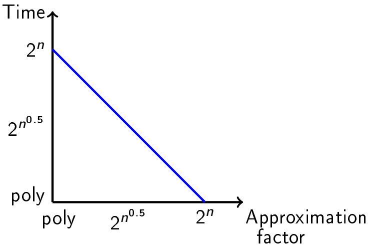
SVP and CVP in general lattices are conjectured to be hard to solve both quantumly and classically $\Rightarrow$ used in cryptography

# 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

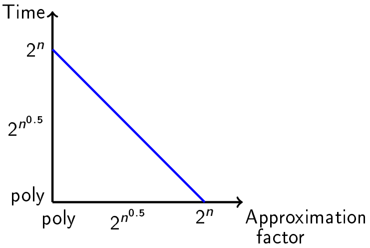


# Complexity of SVP/CVP

## 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




**Applications**

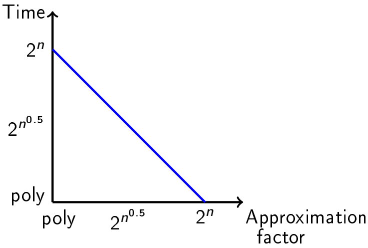
$\boxed{\text{LWE / SIS}} \longrightarrow \boxed{\text{SIVP}}$

# 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




**Applications**

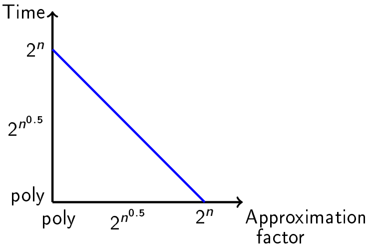
LWE / SIS $\rightleftarrows$ SIVP

# 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




**Applications**

$\boxed{\text{LWE / SIS}} \rightleftarrows \boxed{\text{SIVP}}$

- hash functions
- encryption
- identity based encryption
- fully homomorphic encryption
- ...

# Structured lattices

**Difficulty**

Schemes using LWE/SIS are not efficient

# Structured lattices

## Difficulty

Schemes using LWE/SIS are not efficient

Improve efficiency using structured lattices, e.g.

$$M = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ -a_n & a_1 & \cdots & a_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ -a_2 & -a_3 & \dots & a_1 \end{pmatrix}$$

# Structured lattices

## Difficulty

Schemes using LWE/SIS are not efficient

Improve efficiency using structured lattices, e.g.

$$M = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ -a_n & a_1 & \cdots & a_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ -a_2 & -a_3 & \ldots & a_1 \end{pmatrix}$$

$\Rightarrow$ this is an ideal lattice

# Structured lattices

## Difficulty

Schemes using LWE/SIS are not efficient

Improve efficiency using structured lattices, e.g.

$$M = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ -a_n & a_1 & \cdots & a_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ -a_2 & -a_3 & \dots & a_1 \end{pmatrix}$$

$\Rightarrow$ this is an ideal lattice

Structured analogues of LWE/SIS: $\boxed{\text{Ring LWE, Ring SIS}} \rightarrow \boxed{\text{SVP in ideal lattices}}$

# Structured lattices

## Difficulty

Schemes using LWE/SIS are not efficient

Improve efficiency using structured lattices, e.g.

$$M = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ -a_n & a_1 & \cdots & a_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ -a_2 & -a_3 & \dots & a_1 \end{pmatrix}$$

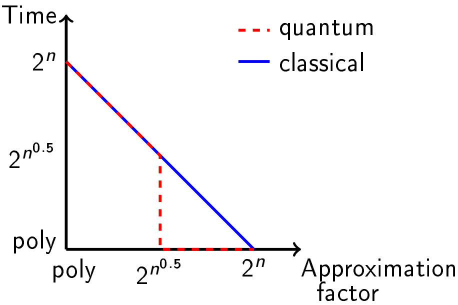
$\Rightarrow$ this is an ideal lattice

Structured analogues of LWE/SIS: $\boxed{\text{Ring LWE, Ring SIS}} \longrightarrow \boxed{\text{SVP in ideal lattices}}$

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

# SVP in ideal lattices

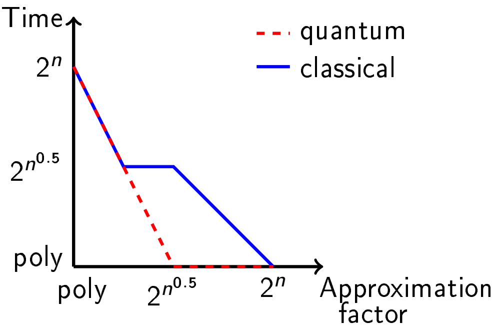
[CDW17]: Better than BKZ in the quantum setting




- 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




(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 general lattices

# Impact

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

# Impact

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

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

# First definitions

**Notation**

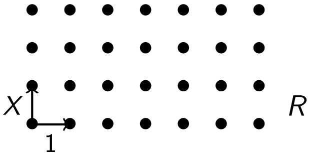
$R = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^k$

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

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

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

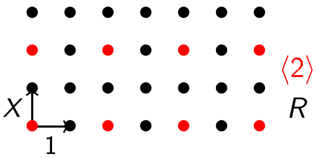


# Why is $\langle g \rangle$ a lattice?

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

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

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



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

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

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

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

**Basis:** $g, gX, gX^2, \ldots, gX^{n-1}$

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

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

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

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

**Basis:** $g, gX, gX^2, \ldots, gX^{n-1}$

$$\text{i.e.,} \quad \begin{pmatrix} g_0 & & & \\ g_1 & & & \\ \vdots & & & \\ g_{n-1} & & & \end{pmatrix}$$

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

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

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

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

**Basis:** $g, gX, gX^2, \ldots, gX^{n-1}$

$$\text{i.e., } \begin{pmatrix} g_0 & -g_{n-1} & & \\ g_1 & g_0 & & \\ \vdots & \vdots & & \\ g_{n-1} & g_{n-2} & & \end{pmatrix}$$

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

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

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

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

**Basis:** $g, gX, gX^2, \ldots, gX^{n-1}$

$$\text{i.e.,} \quad \begin{pmatrix} g_0 & -g_{n-1} & \cdots & -g_1 \\ g_1 & g_0 & \cdots & -g_2 \\ \vdots & \vdots & \ddots & \vdots \\ g_{n-1} & g_{n-2} & \cdots & g_0 \end{pmatrix}$$

# Objective of this talk

## Objective

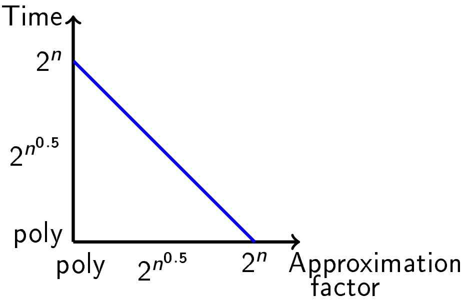
Given a basis of a principal ideal $\langle g \rangle$ and $\alpha \in (0, 1]$,
Find $r \in \langle g \rangle \setminus \{0\}$ such that $\|r\| \leq 2^{\widetilde{O}(n^\alpha)} \cdot \lambda_1$.

# 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?



# Outline of the talk

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

**Idea**

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

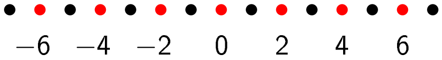
[CGS14]: P. Campbell, M. Groves, and D. Shepherd. Soliloquy: A cautionary tale.

# 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.



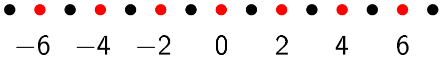

[CGS14]: P. Campbell, M. Groves, and D. Shepherd. Soliloquy: A cautionary tale.

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

**Idea**

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

**If 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

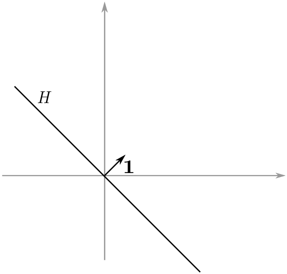
---

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

# The Log space

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

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



# The Log space

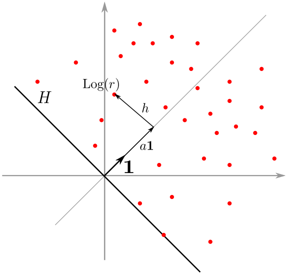
Log : $R \to \mathbb{R}^n$ (somehow generalising log to $R$)

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

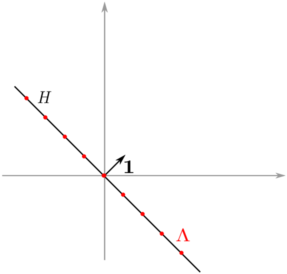
$\mathrm{Log} : R \to \mathbb{R}^n$ (somehow generalising log to $R$)

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



# The Log space

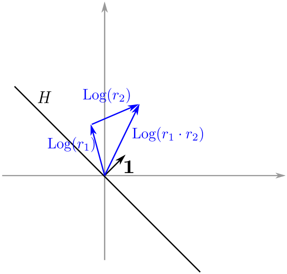
$\text{Log} : R \to \mathbb{R}^n$ (somehow generalising log to $R$)

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



# The Log space

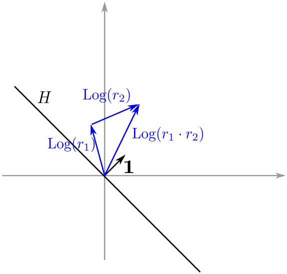
$\text{Log} : R \to \mathbb{R}^n$ (somehow generalising log to $R$)

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

## Properties

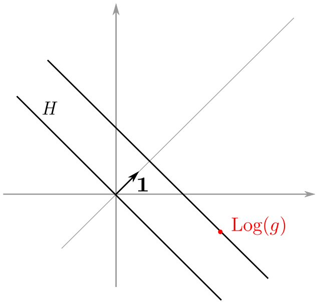
$\text{Log}\, r = h + a\mathbf{1}$, with $h \in H$

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



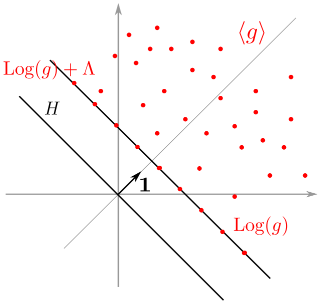
# The CDPR algorithm

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



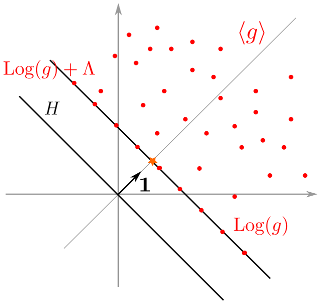
# The CDPR algorithm

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



# The CDPR algorithm

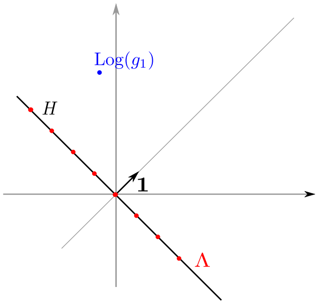
What does $\mathrm{Log}\langle g \rangle$ look like?



# The CDPR algorithm

**The CDPR Algorithm:**

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



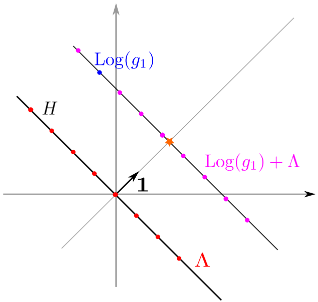

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

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



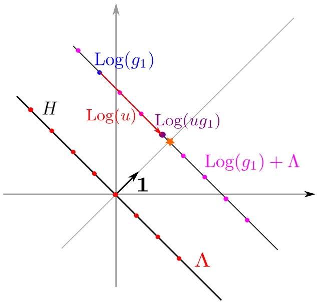

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

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




---

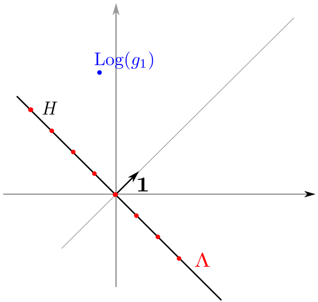
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

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



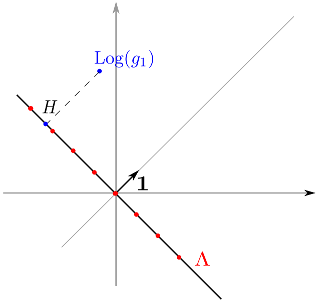

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

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



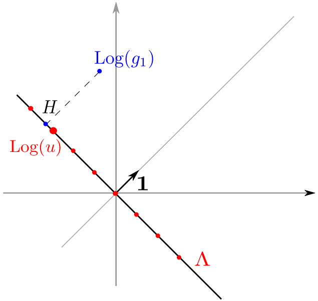

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $\Lambda$



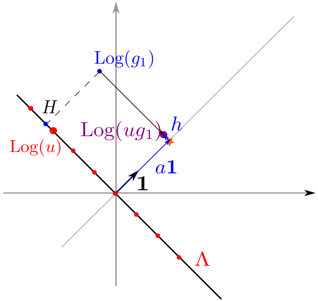

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $\Lambda$



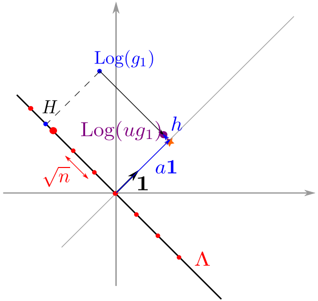

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $\Lambda$
  - Good basis of $\Lambda$
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$



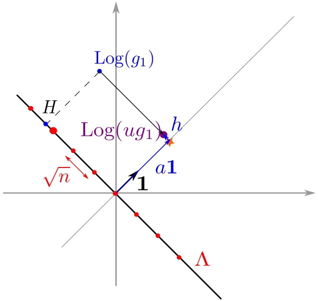

[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $\Lambda$
  - Good basis of $\Lambda$
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$




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

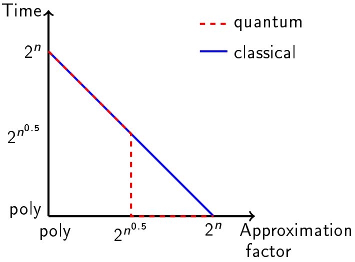
[BS16]: J.F. Biasse, F. Song. Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, SODA.

[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 $\Lambda$
  - Good basis of $\Lambda$
    $\Rightarrow$ CVP in poly time
    $\Rightarrow \|h\| \leq \widetilde{O}(\sqrt{n})$




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

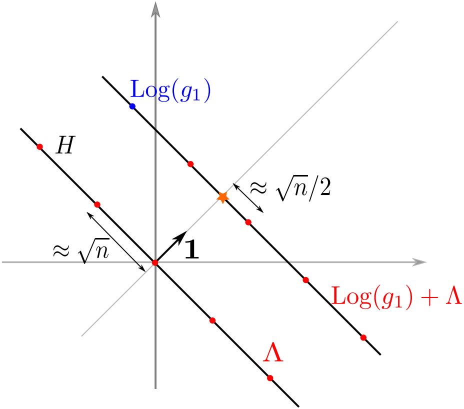
---

[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.

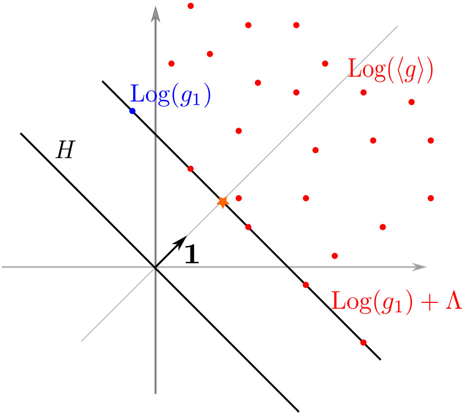
# Outline of the talk

# Idea



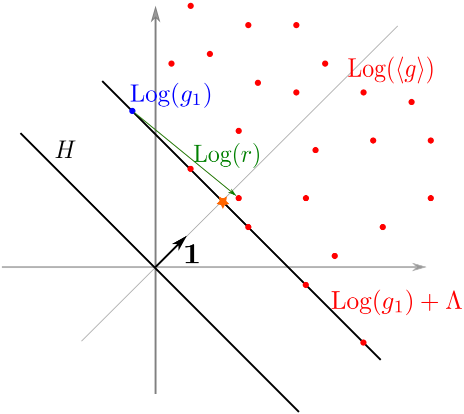
$\mathrm{Log}(g_1)$
$H$
$\approx \sqrt{n}/2$
$\approx \sqrt{n}$
$\mathbf{1}$
$\mathrm{Log}(g_1) + \Lambda$
$\Lambda$

# Idea



$\mathrm{Log}(\langle g \rangle)$
$\mathrm{Log}(g_1)$
$H$
$\mathbf{1}$
$\mathrm{Log}(g_1) + \Lambda$

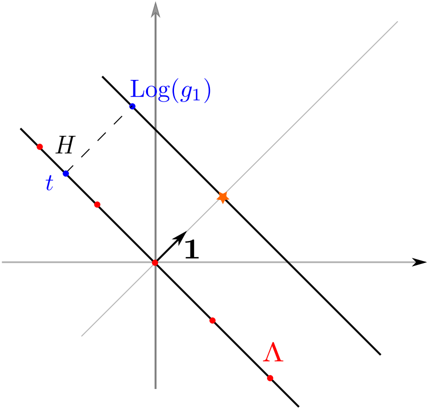
# Idea




## Important

$\mathrm{Log}\, r = h + a\mathbf{1}$ with $a$ small (and $h \in H$).

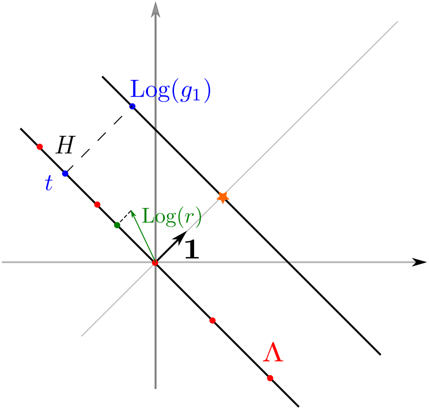
# Idea




## Important

$\mathrm{Log}\, r = h + a\mathbf{1}$ with $a$ small (and $h \in H$).

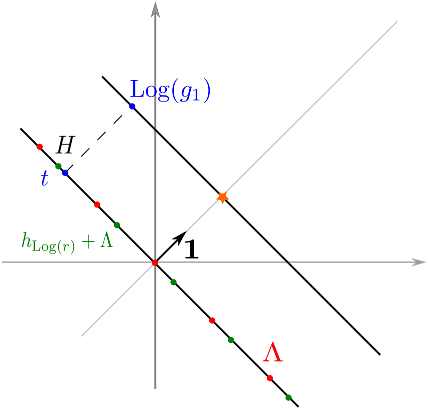
# Idea




**Important**

$\mathrm{Log}\, r = h + a\mathbf{1}$ with $a$ small (and $h \in H$).

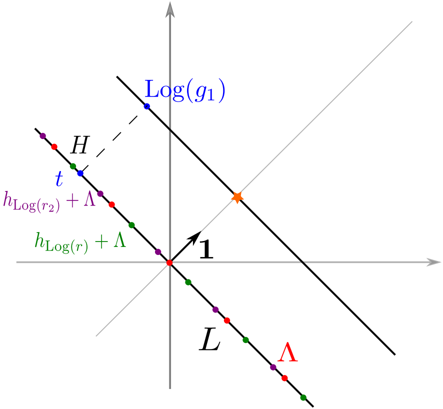
# Idea




## Important

Log $r = h + a\mathbf{1}$ with $a$ small (and $h \in H$).

# Idea




## Important

$\operatorname{Log} r = h + a\mathbf{1}$ with $a$ small (and $h \in H$).

# The lattice $L$

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

# The lattice $L$

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

# The lattice $L$

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

## 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 $\|t - s\| = O(1)$)

# Algorithm

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$'

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$'

Compute the lattice $L$

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$'

Compute the lattice $L$

Compute $g_1$ a generator of $\langle g \rangle$,
$\Rightarrow$ let $t = (h_{\mathrm{Log}(g_1)}, 0, \cdots, 0)^T$

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$'

Compute the lattice $L$

Compute $g_1$ a generator of $\langle g \rangle$,
$\Rightarrow$ let $t = (h_{\mathrm{Log}(g_1)}, 0, \cdots, 0)^T$

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

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$'

Compute the lattice $L$

Compute $g_1$ a generator of $\langle g \rangle$,
$\Rightarrow$ let $t = (h_{\mathrm{Log}(g_1)}, 0, \cdots, 0)^T$

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 $s = h_{\mathrm{Log}\, r}$ for some $r \in R$

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$'

Compute the lattice $L$

Compute $g_1$ a generator of $\langle g \rangle$,
$\Rightarrow$ let $t = (h_{\mathrm{Log}(g_1)}, 0, \cdots, 0)^T$

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 $s = h_{\mathrm{Log}\, r}$ for some $r \in R$

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

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$' $\quad$ $\text{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Compute the lattice $L$

Compute $g_1$ a generator of $\langle g \rangle$, $\quad$ $\text{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$
$\Rightarrow$ let $t = (h_{\text{Log}(g_1)}, 0, \cdots, 0)^T$

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 $s = h_{\text{Log}\, r}$ for some $r \in R$

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

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$' $\quad \mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Compute the lattice $L$ $\quad \mathrm{poly}(n)$

Compute $g_1$ a generator of $\langle g \rangle$, $\quad \mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$
$\Rightarrow$ let $t = (h_{\mathrm{Log}(g_1)}, 0, \cdots, 0)^T$

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 $s = h_{\mathrm{Log}\, r}$ for some $r \in R$ $\quad \mathrm{poly}(n)$

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

# Algorithm

Compute $r_1, \cdots, r_n$ with small '$a$' $\quad \mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

Compute the lattice $L$ $\quad \mathrm{poly}(n)$

Compute $g_1$ a generator of $\langle g \rangle$,
$\Rightarrow$ let $t = (h_{\mathrm{Log}(g_1)}, 0, \cdots, 0)^T$ $\quad \mathrm{poly}(n)$ / $2^{\widetilde{O}(\sqrt{n})}$

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

Write $s = h_{\mathrm{Log}\, r}$ for some $r \in R$ $\quad \mathrm{poly}(n)$

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

# How to solve CVP in $L$?

| CDPR | This work |
|---|---|
| Good basis of $\Lambda$ | No good basis of $L$ known |

# How to solve CVP in $L$?

| CDPR | This work |
| --- | --- |
| Good basis of $\Lambda$ | 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 $\Lambda$ | 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 $\Lambda$ | 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)

---

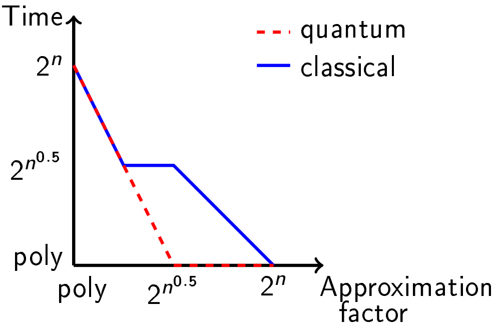
[Laa16] T. Laarhoven. Finding closest lattice vectors using approximate Voronoi cells. SAC.

[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

# Outline of the talk

# Extensions

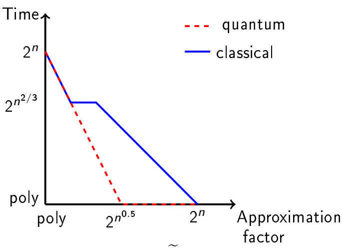
We can extend the algorithm to

- Non-principal ideals

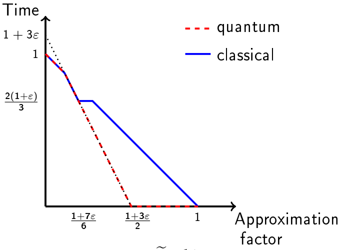
# Extensions

We can extend the algorithm to

- Non-principal ideals
- All number fields




$\log|\Delta| = \widetilde{O}(n)$




$\log|\Delta| = \widetilde{O}(n^{1+\varepsilon})$

# Extension to any ideal

## Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

# Extension to any ideal

### Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

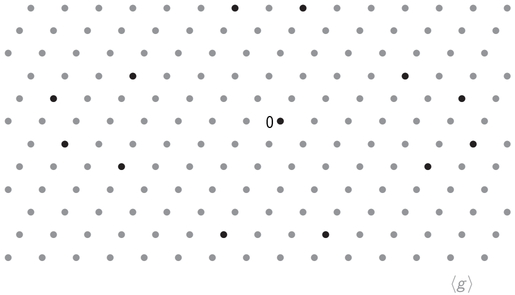
[CDPR]: find the smallest generator of a principal ideal

# Extension to any ideal

## Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

[CDPR]: find the smallest generator of a principal ideal

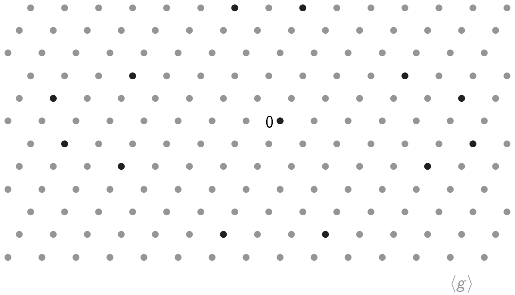


# Extension to any ideal

## Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

[CDPR]: find the smallest generator of a principal ideal




What we did

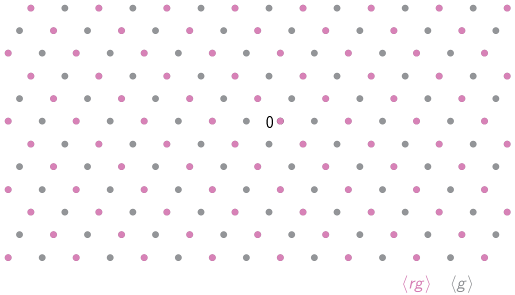
- All generators • are somehow large

# Extension to any ideal

## Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

[CDPR]: find the smallest generator of a principal ideal




What we did

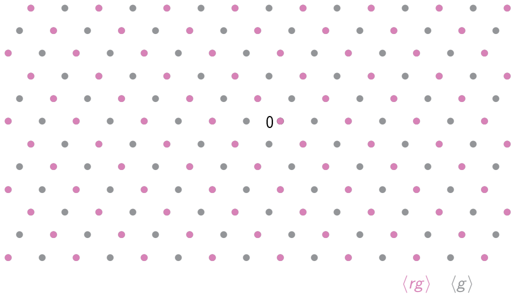
- All generators • are somehow large
- Multiply by some small $r$
  - $\langle rg \rangle$ sublattice of $\langle g \rangle$

# Extension to any ideal

## Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

[CDPR]: find the smallest generator of a principal ideal




What we did

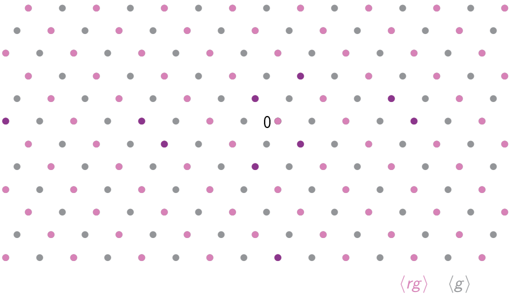
- All generators • are somehow large
- Multiply by some small $r$
  - $\langle rg \rangle$ sublattice of $\langle g \rangle$
  - not much sparser

# Extension to any ideal

## Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

[CDPR]: find the smallest generator of a principal ideal




What we did

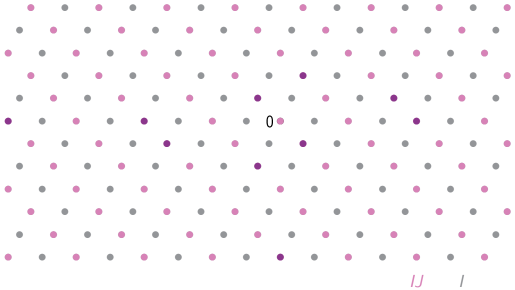
- All generators • are somehow large
- Multiply by some small $r$
  - $\langle rg \rangle$ sublattice of $\langle g \rangle$
  - not much sparser
  - with a small generator •

# Extension to any ideal

## Ideal

An ideal is $I = \{ar_1 + br_2,\ r_1, r_2 \in R\}$ for some $a, b \in R$
A principal ideal is $\langle g \rangle = \{gr, r \in R\}$ for some $g \in R$.

[CDPR]: find the smallest generator of a principal ideal




Extension to any ideal

- $I$ has no generator (not principal)
- Multiply by some small ideal $J$
  - $IJ$ sublattice of $I$
  - not much sparser
  - principal
  - with a small generator •

# Perspectives and open questions

**Reminder:** This is a theoretical result (no practical applications)

[BBV+17] J. Bauch, D. J. Bernstein, H. de Valence, T. Lange, C. van Vredendaal. Short generators without quantum computers: the case of multiquadratics. Eurocrypt.

# Perspectives and open questions

**Reminder:** This is a theoretical result (no practical applications)

**Open problems:**

- Remove/prove the heuristics?
  - Stephens-Davidowitz proved Laarhoven's algo

[BBV+17] J. Bauch, D. J. Bernstein, H. de Valence, T. Lange, C. van Vredendaal. Short generators without quantum computers: the case of multiquadratics. Eurocrypt.

# Perspectives and open questions

**Reminder:** This is a theoretical result (no practical applications)

**Open problems:**

- Remove/prove the heuristics?
  - Stephens-Davidowitz proved Laarhoven's algo
- Improve the algorithm for specific rings?
  - e.g. multiquadratics [BBV+17]

[BBV+17] J. Bauch, D. J. Bernstein, H. de Valence, T. Lange, C. van Vredendaal. Short generators without quantum computers: the case of multiquadratics. Eurocrypt.

# Perspectives and open questions

**Reminder:** This is a theoretical result (no practical applications)

**Open problems:**

- Remove/prove the heuristics?
  - Stephens-Davidowitz proved Laarhoven's algo
- Improve the algorithm for specific rings?
  - e.g. multiquadratics [BBV+17]
- Generalize to module SVP?

---

[BBV+17] J. Bauch, D. J. Bernstein, H. de Valence, T. Lange, C. van Vredendaal. Short generators without quantum computers: the case of multiquadratics. Eurocrypt.

# Perspectives and open questions

**Reminder:** This is a theoretical result (no practical applications)

**Open problems:**

- Remove/prove the heuristics?
  - Stephens-Davidowitz proved Laarhoven's algo
- Improve the algorithm for specific rings?
  - e.g. multiquadratics [BBV+17]
- Generalize to module SVP?

Questions?

[BBV+17] J. Bauch, D. J. Bernstein, H. de Valence, T. Lange, C. van Vredendaal. Short generators without quantum computers: the case of multiquadratics. Eurocrypt.