Hardness of Structured Lattice Problems for Post-Quantum Cryptography

Under the supervision of Damien Stehlé and Guillaume Hanrot

Joël Felderhoff

INRIA Lyon, ENS de Lyon

26/11/2024

Introduction: why study structured lattice problems?

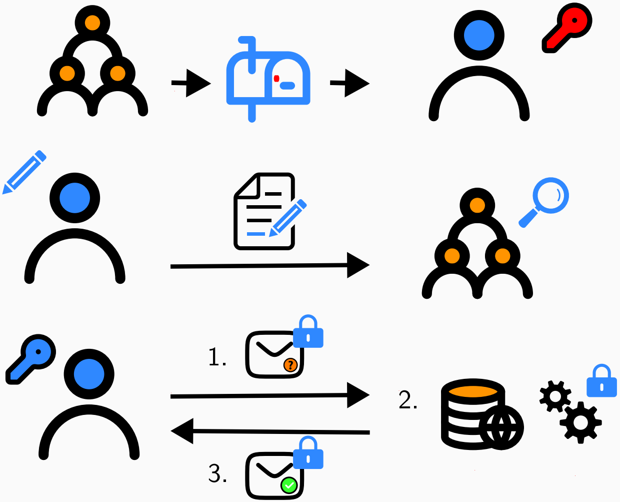
Communication Security

Some Example of Protocols

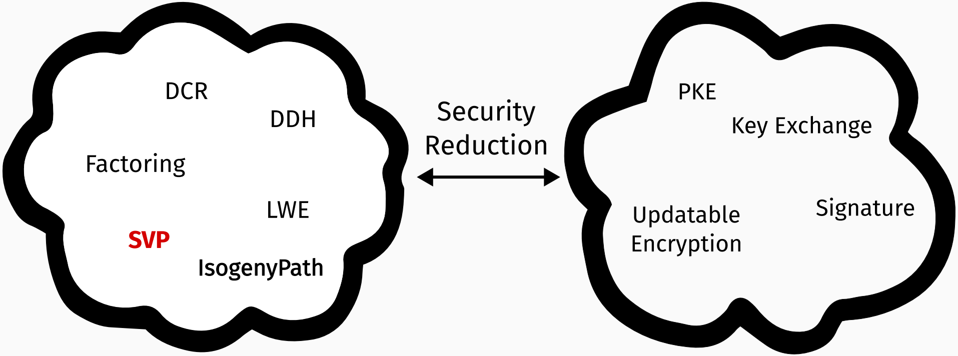
Public Key Encryption

Signature

Homomorphic Encryption



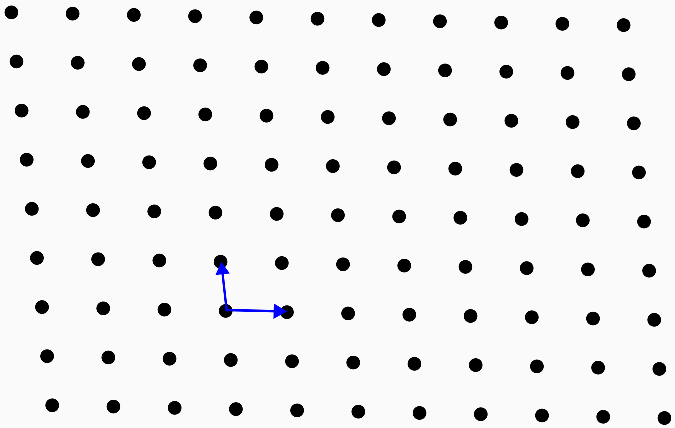
Security proof and problem hardness



Well known
Easy to study

Useful in practice

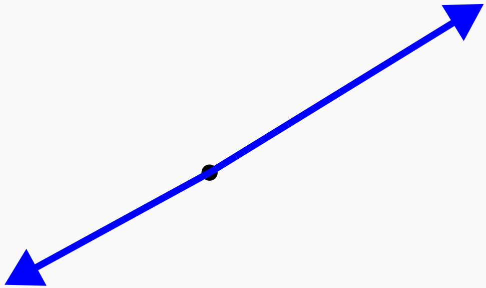
Our Mathematical Object of Choice: Lattices



Lattice spanned by $\boldsymbol{B} \in \mathbb{Z}^{n \times n}$:

$$\mathcal{L}(\boldsymbol{B}) = \{\boldsymbol{B} \cdot \boldsymbol{x},\ \boldsymbol{x} \in \mathbb{Z}^n\}.$$

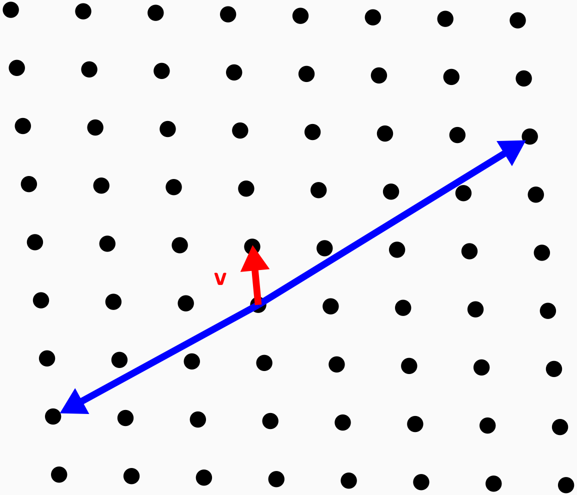
The Hard Problem for Lattices: Finding a Short Vector



Shortest Vector Problem (SVP)

Given $\boldsymbol{B}$, find a shortest non-zero vector $\boldsymbol{v}$ in the lattice spanned by $\boldsymbol{B}$.

The Hard Problem for Lattices: Finding a Short Vector

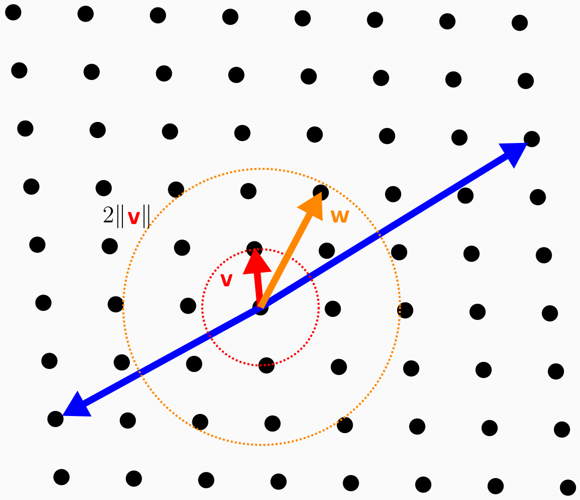


Shortest Vector Problem (SVP)

Given $\boldsymbol{B}$, find a shortest non-zero vector $\boldsymbol{v}$ in the lattice spanned by $\boldsymbol{B}$.

In dimension n:
Finding $\boldsymbol{v}$: $\sim 2^{O(n)}$ op.

The Hard Problem for Lattices: Finding a Short Vector

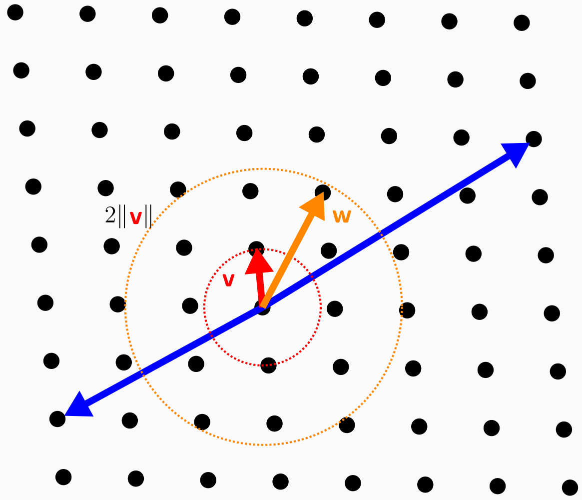


Approx SVP (SVP_γ)

Given $\boldsymbol{B}$, find a short non-zero $\boldsymbol{w}$ in the lattice spanned by $\boldsymbol{B}$ with $\|\boldsymbol{w}\| \leq \gamma \cdot \|\boldsymbol{v}\|$.

In dimension n:
Finding $\boldsymbol{v}$: $\sim 2^{O(n)}$ op.
Finding $\boldsymbol{w}$: $\sim 2^{O(n)}/\gamma$ op.
Seems hard **even with quantum computers**.

The Hard Problem for Lattices: Finding a Short Vector



Approx SVP (SVP$_\gamma$)

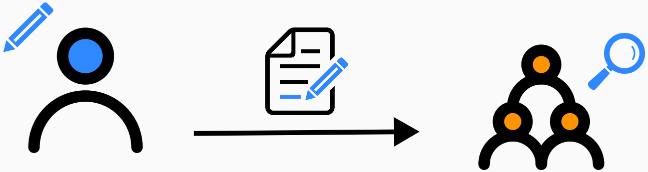
Given $\boldsymbol{B}$, find a short non-zero $\boldsymbol{w}$ in the lattice spanned by $\boldsymbol{B}$ with $\|\boldsymbol{w}\| \leq \gamma \cdot \|\boldsymbol{v}\|$.

In dimension n:
Finding $\boldsymbol{v}$: $\sim 2^{O(n)}$ op.
Finding $\boldsymbol{w}$: $\sim 2^{O(n)}/\gamma$ op.
Seems hard **even with quantum computers**.

In cryptography, typically
$n \simeq 1000$, $\gamma = \text{poly}(n)$.

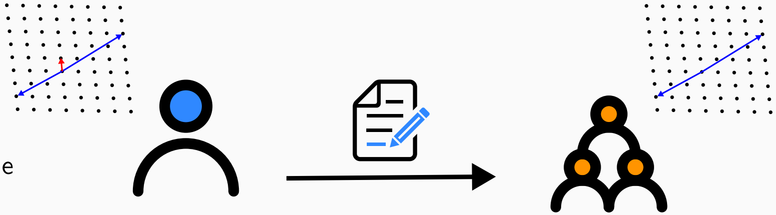
Structured Lattices: Motivation

Signature scheme



Structured Lattices: Motivation

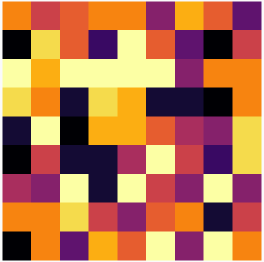
Signature scheme



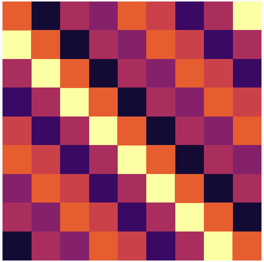
Using any matrix $\boldsymbol{B} \in \mathbb{Z}^{n \times n}$: n^2 coefficients, long running-time, memory inefficient.

Structured Lattices

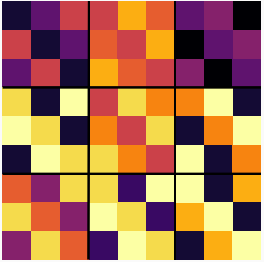
Idea: use matrices with structure (e.g. from algebraic number theory).
$\rightarrow$ **Module Lattices**.



Unstructured



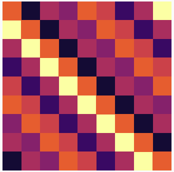
Rank 1
Ideal Lattices



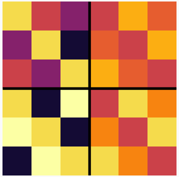
Rank 3
Module lattice

New Lattices, New (easier) Problems

Attacks exist

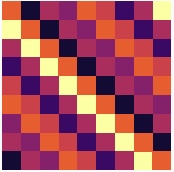
SVP for B =  → Ideal HSVP (idHSVP)

???

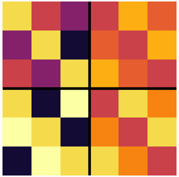
SVP for B =  → Module SVP (modSVP)

New Lattices, New (easier) Problems

Attacks exist

SVP for B =  → Ideal HSVP (idHSVP)

???

SVP for B =  → Module SVP (modSVP)

Cramer, Ducas, Peikert, and Regev. Recovering short generators of principal ideals in cyclotomic rings. EUROCRYPT, 2016.
Cramer, Ducas, and Wesolowski. Short Stickelberger class relations and application to Ideal-SVP. EUROCRYPT, 2017.
Pellet-Mary, Hanrot, and Stehlé. Approx-SVP in ideal lattices with pre-processing. EUROCRYPT, 2019.

What do I mean, “easier”?

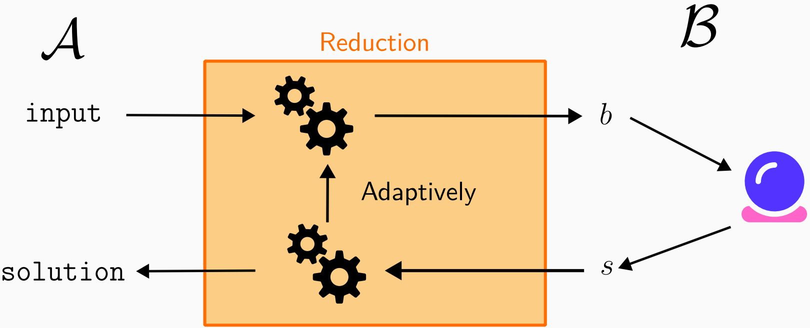
$$\text{Hardness}(\mathcal{A}) \leq \text{Hardness}(\mathcal{B})$$

What do I mean, "easier"?

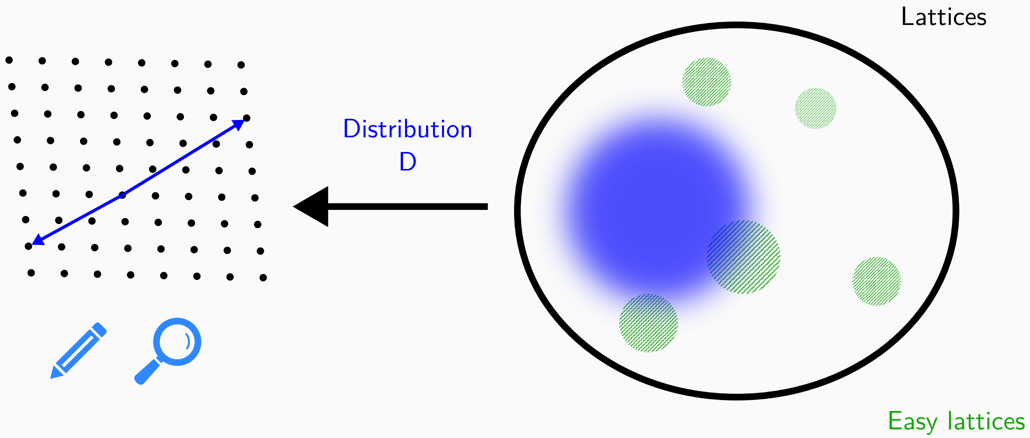
$$\text{Hardness}(\mathcal{A}) \leq \text{Hardness}(\mathcal{B}) \equiv \text{“If someone solves } \mathcal{B}\text{, they can solve } \mathcal{A}\text{”.}$$

What do I mean, "easier"?

$\text{Hardness}(\mathcal{A}) \leq \text{Hardness}(\mathcal{B}) \equiv$ "If someone solves $\mathcal{B}$, they can solve $\mathcal{A}$".

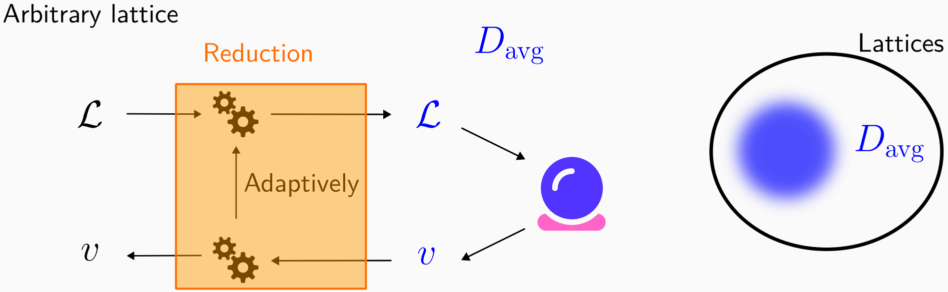


In this presentation: Average-case problems



How do we avoid easy lattices?

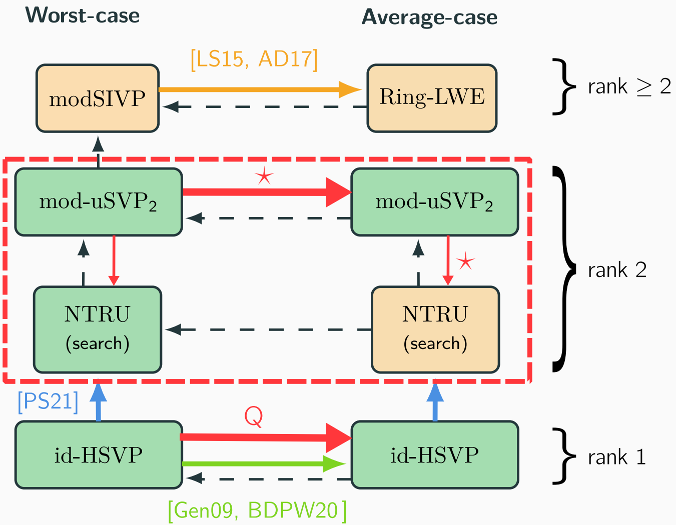
Average-case reductions



Oracle on D_{avg} strong enough to break any lattice

$\rightarrow$ D_{avg} avoids easy lattices

What I did during my Phd



[LS15] Langlois, Stehlé. Worst-case to average-case reductions for module lattices. DCC 2014.

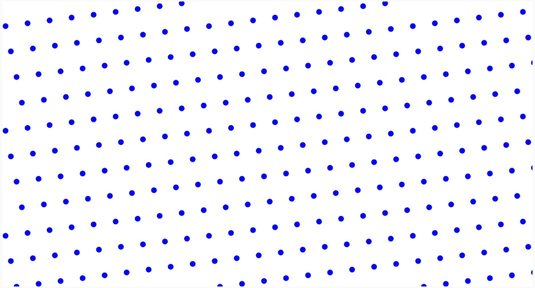
[AD17] Albrecht, Deo. Large Modulus Ring-LWE Module-LWE. ASIACRYPT 2017.

[PS21] Pellet-Mary, Stehlé. On the hardness of the NTRU problem. ASIACRYPT 2021.

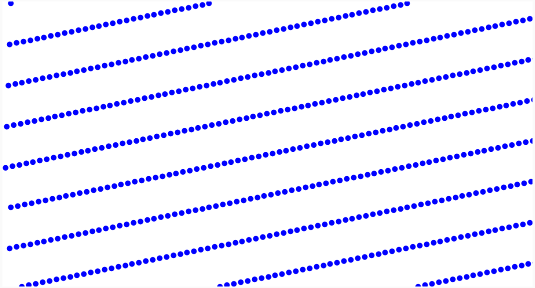
[Gen09] Gentry. A Fully Homomorphic Encryption Scheme. PhD thesis. 2009.

[BDPW20] de Boer, Ducas, Pellet-Mary, Wesolowski. Random self-reducibility of Ideal-SVP via Arakelov random walks. CRYPTO, 2020.

Worst-case to Average-case reduction for mod-uSVP_2



Typical lattice



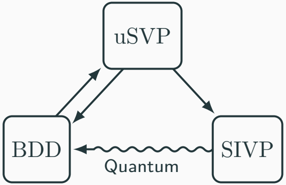
mod-uSVP$_2$ instance

γ-mod-uSVP$_2$

Given a basis $\boldsymbol{B}$ of a module $M \subset \mathcal{O}_K^2$ s.t. $\lambda_1(M) \leq \det(\boldsymbol{B})^{1/(2d)}/\gamma$, find a short non-zero vector in it.

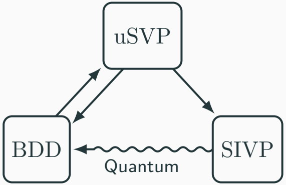
State of the art for $\mathrm{mod\text{-}uSVP}_2$

For $\mathbb{Z}$-lattices

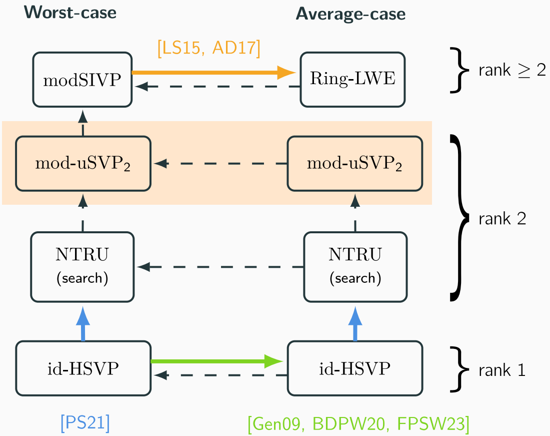


State of the art for $\mathrm{mod\text{-}uSVP}_2$

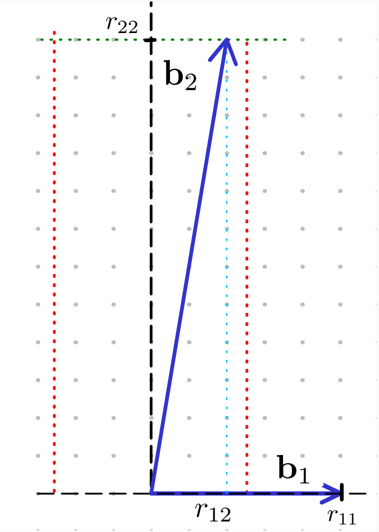
For $\mathbb{Z}$-lattices



For $\mathcal{O}_K$-modules



Anatomy of a $\mathrm{mod\text{-}uSVP}_2$ instance: QR factorization



Any (free) $\mathrm{mod\text{-}uSVP}_2$ instance has a basis

$$\boldsymbol{B} = \boldsymbol{Q} \cdot \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}$$

with $r_{11} \ll r_{22}$, $r_{12} \in \left(\frac{-r_{11}}{2}, \frac{r_{11}}{2}\right)$ and $\boldsymbol{Q}$ orthogonal.

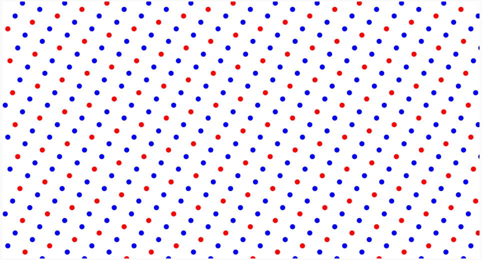
Goal for the randomization:

- Randomize $\boldsymbol{Q}$.
- Randomize r_{11} and r_{22}.
- Randomize r_{12}.

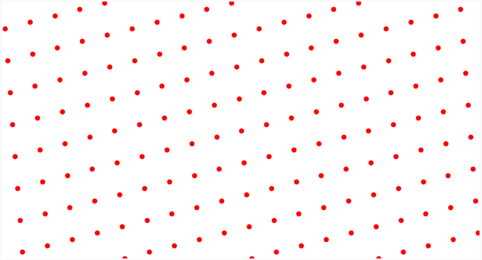
Difficulty: we don't have access to the good basis.

Randomization of r_{11} and r_{22}

We multiply by a scalar: this changes r_{11} and r_{22} but r_{11}/r_{22} is fixed.
Solution: sparsification by a prime p.



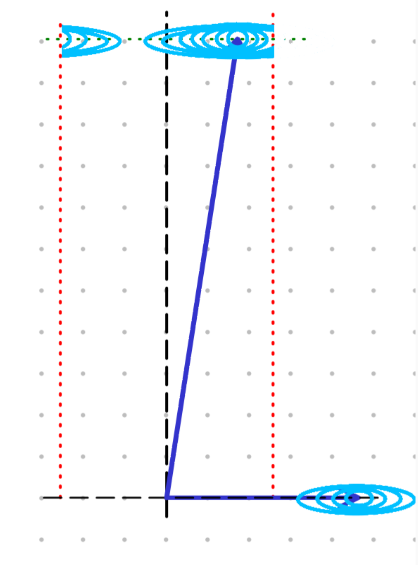
$\Longrightarrow$



Sparsification by p

Only keep 1 every p points. Multiplies r_{11} by p with high probability and leaves r_{22} unchanged.

Randomization of r_{12}



Idea: blur the space with a matrix $\boldsymbol{D}$.

$$\boldsymbol{D} \cdot \boldsymbol{Q} \sim \boldsymbol{D} = \boldsymbol{Q'} \cdot \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}.$$

Then

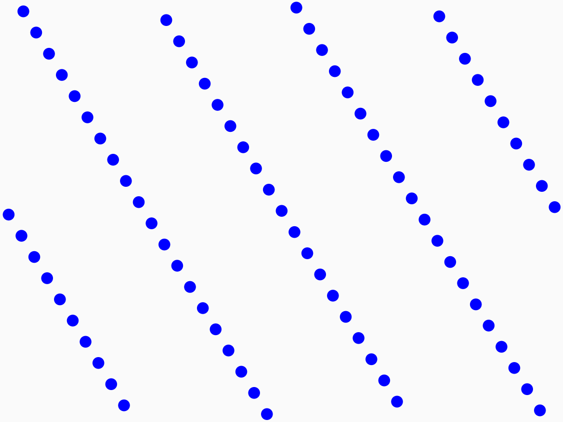
$$\boldsymbol{M'} = \boldsymbol{D} \cdot \boldsymbol{M} \sim \boldsymbol{Q'} \cdot \begin{pmatrix} r'_{11} & r'_{12} \\ 0 & r'_{22} \end{pmatrix}$$

where

$$\begin{aligned} r'_{12} =& (b + a r_{12}) \bmod r'_{11} \\ \approx& \mathrm{Unif}(\mathcal{O}_K \bmod r'_{11}) \end{aligned}$$

when $\boldsymbol{D}$ is a Gaussian.

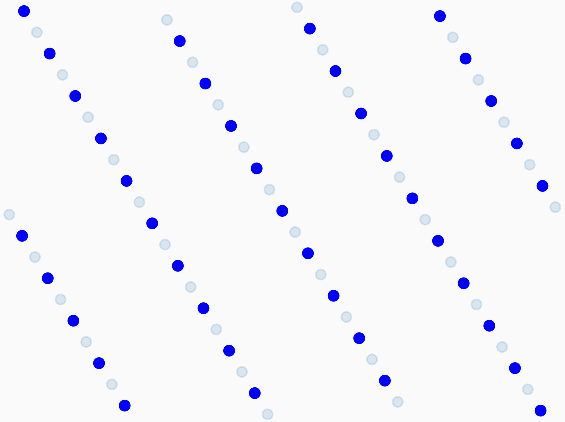
Visualization of the reduction



Randomization

Input: M_{input}.

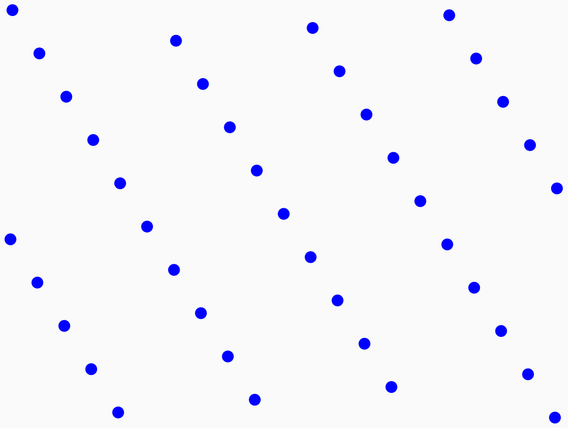
Visualization of the reduction



Randomization

Input: M_{input}.
Sparsification:
$M_2 := M_{input} \cdot S$.

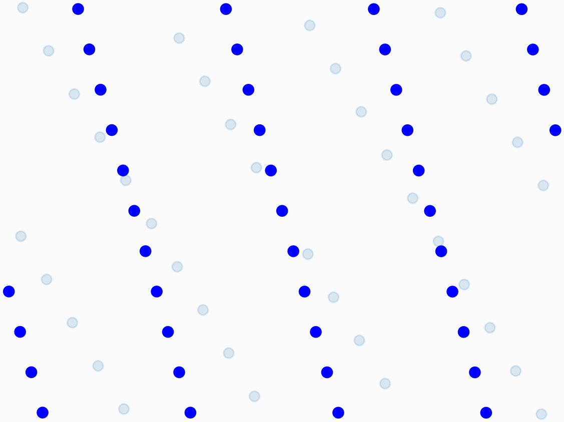
Visualization of the reduction



Randomization

Input: M_{input}.
Sparsification:
$M_2 := M_{input} \cdot S$.

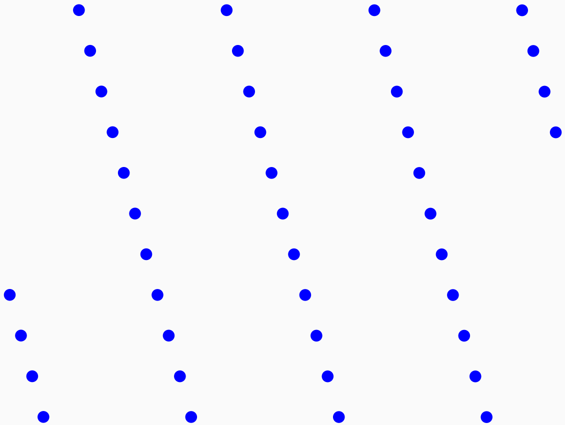
Visualization of the reduction



Randomization

Input: $\boldsymbol{M}_{input}$.
Sparsification:
$\boldsymbol{M}_2 := \boldsymbol{M}_{input} \cdot \boldsymbol{S}$.
Gaussian:
$\boldsymbol{M}_{random} := \boldsymbol{G} \cdot \boldsymbol{M}_2$

Visualization of the reduction



Randomization

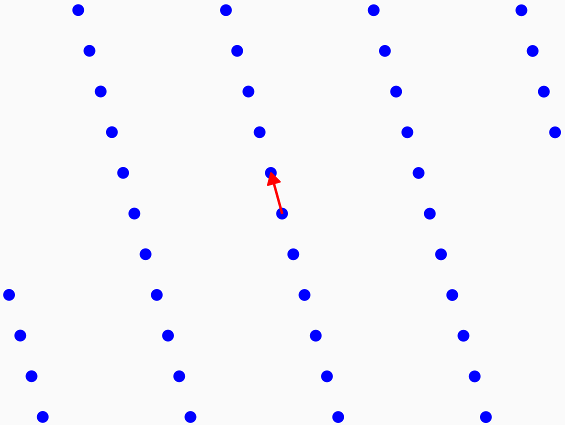
Input: $\boldsymbol{M}_{input}$.
Sparsification:
$\boldsymbol{M}_2 := \boldsymbol{M}_{input} \cdot \boldsymbol{S}$.
Gaussian:
$\boldsymbol{M}_{random} := \boldsymbol{G} \cdot \boldsymbol{M}_2$

Visualization of the reduction

Randomization

Input: $\boldsymbol{M}_{input}$.
Sparsification:
$\boldsymbol{M}_2 := \boldsymbol{M}_{input} \cdot \boldsymbol{S}$.
Gaussian:
$\boldsymbol{M}_{random} := \boldsymbol{G} \cdot \boldsymbol{M}_2$
Magic* happens.

Visualization of the reduction



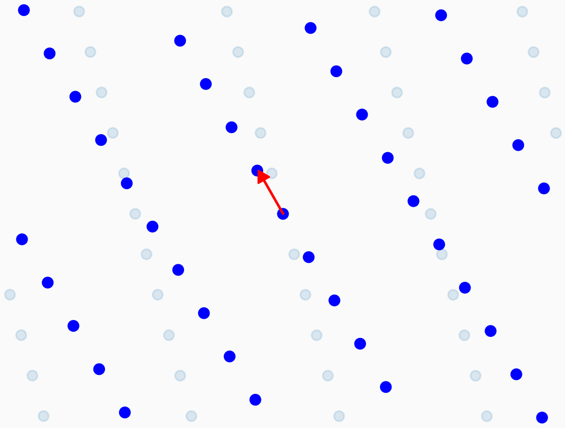
Randomization

Input: $\boldsymbol{M}_{input}$.
Sparsification:
$\boldsymbol{M}_2 := \boldsymbol{M}_{input} \cdot \boldsymbol{S}$.
Gaussian:
$\boldsymbol{M}_{random} := \boldsymbol{G} \cdot \boldsymbol{M}_2$
Magic* happens.

Retrieving short vector in $\boldsymbol{M}_{input}$

Oracle:
$\mathcal{O}(\boldsymbol{M}_{random}) \to \boldsymbol{v}_1 \in \boldsymbol{M}_{random}$.

Visualization of the reduction



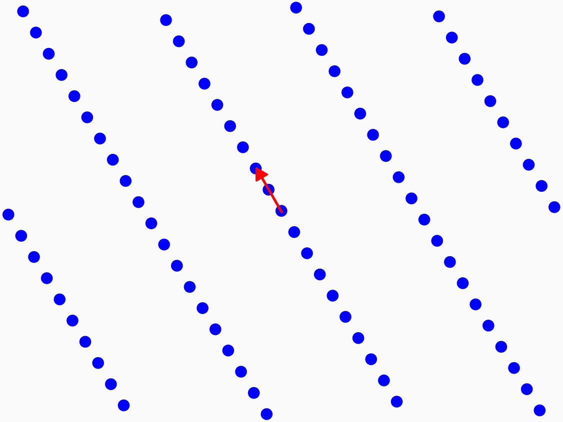
Randomization

Input: M_{input}.
Sparsification:
$M_2 := M_{input} \cdot S$.
Gaussian:
$M_{random} := G \cdot M_2$
Magic* happens.

Retrieving short vector in M_{input}

Oracle:
$\mathcal{O}(M_{random}) \to v_1 \in M_{random}$.
Gaussian^{-1}:
$v_2 = G^{-1} \cdot v_1 \in M_2$.

Visualization of the reduction



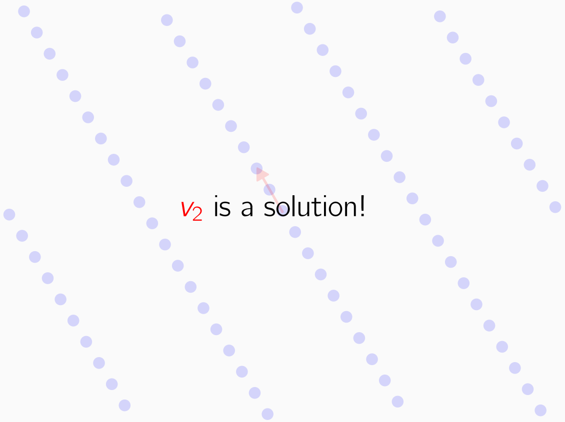
Randomization

Input: $\boldsymbol{M}_{input}$.
Sparsification:
$\boldsymbol{M}_2 := \boldsymbol{M}_{input} \cdot \boldsymbol{S}$.
Gaussian:
$\boldsymbol{M}_{random} := \boldsymbol{G} \cdot \boldsymbol{M}_2$
Magic* happens.

Retrieving short vector in $\boldsymbol{M}_{input}$

Oracle:
$\mathcal{O}(\boldsymbol{M}_{random}) \to \boldsymbol{v}_1 \in \boldsymbol{M}_{random}$.
Gaussian^{-1}:
$\boldsymbol{v}_2 = \boldsymbol{G}^{-1} \cdot \boldsymbol{v}_1 \in \boldsymbol{M}_2$.
Sparsification^{-1}:
$\boldsymbol{v}_2 \in \boldsymbol{M}_2 \subset \boldsymbol{M}_{input}$

Visualization of the reduction



Randomization

Input: $\boldsymbol{M}_{input}$.
Sparsification:
$\boldsymbol{M}_2 := \boldsymbol{M}_{input} \cdot \boldsymbol{S}$.
Gaussian:
$\boldsymbol{M}_{random} := \boldsymbol{G} \cdot \boldsymbol{M}_2$
Magic* happens.

Retrieving short vector in $\boldsymbol{M}_{input}$

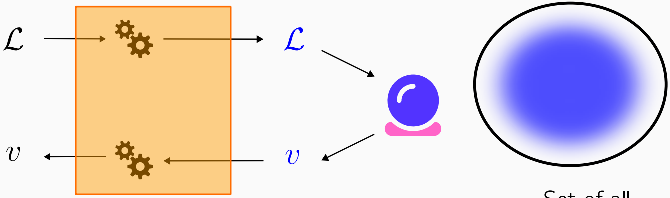
Oracle:
$\mathcal{O}(\boldsymbol{M}_{random}) \to \boldsymbol{v}_1 \in \boldsymbol{M}_{random}$.
Gaussian^{-1}:
$\boldsymbol{v}_2 = \boldsymbol{G}^{-1} \cdot \boldsymbol{v}_1 \in \boldsymbol{M}_2$.
Sparsification^{-1}:
$\boldsymbol{v}_2 \in \boldsymbol{M}_2 \subset \boldsymbol{M}_{input} \to \boldsymbol{v}_2$ is a solution!

Summary: Simplified Statement

Arbitrary
γ-mod-uSVP$_2$
lattice

Randomized
γ-mod-uSVP$_2$
lattice

D_{avg}



The last slides

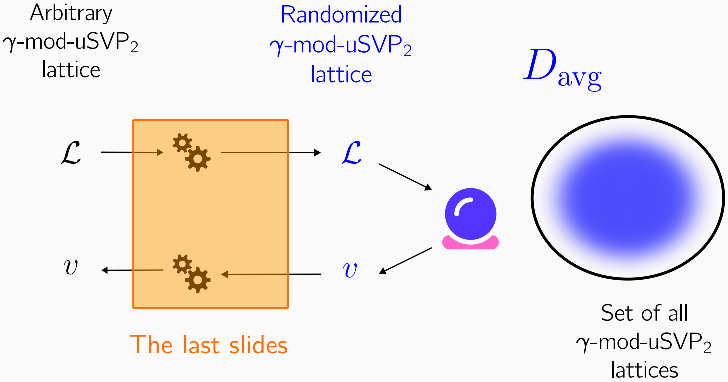
Set of all
γ-mod-uSVP$_2$
lattices

$$D_{\text{avg}}$$

$$Q \cdot \left[\begin{array}{cc} \frac{1}{\gamma}\cdot J_1 & \gamma \cdot J_2 \\ \left(\begin{matrix} 1 \\ 0 \end{matrix}\right. & \left.\begin{matrix} x \\ 1 \end{matrix}\right) \end{array}\right]$$

$$\begin{cases} J_1, J_2 \text{ uniform norm-1;} \\ x \text{ uniform mod } J_1/\gamma; \\ Q \text{ uniform orthogonal.} \end{cases}$$

Summary: Simplified Statement



$$D_{\text{avg}}$$

$$\boldsymbol{Q} \cdot \begin{bmatrix} \frac{1}{\gamma} \cdot J_1 & \gamma \cdot J_2 \\ \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \end{bmatrix}$$

$$\begin{cases} J_1, J_2 \text{ uniform norm-1;} \\ x \text{ uniform mod } J_1/\gamma; \\ \boldsymbol{Q} \text{ uniform orthogonal.} \end{cases}$$

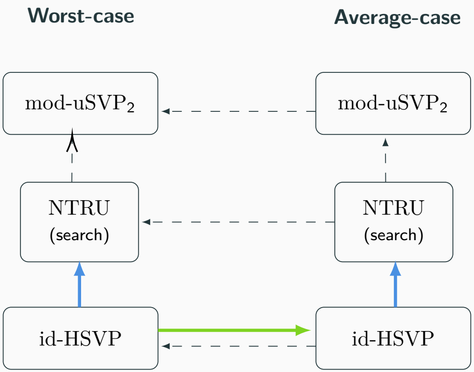
Theorem

Solving $\gamma\text{-mod-uSVP}_2$ *reduces to solving* mod-uSVP_2 *for a lattice sampled from* D_{avg} *w.h.p.*

Under the Rug

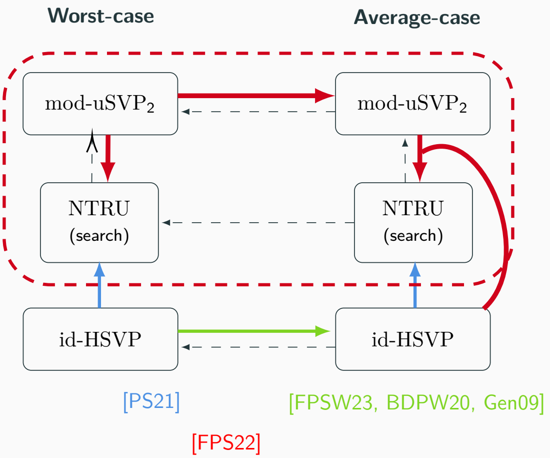
- We are working with number fields all along.
- Non-free modules?
- How to round our module to have integers?
- Change in the approximation factor.
- Running time.
- Randomizing is not exact.

Contributions on $\mathrm{mod\text{-}uSVP}_2$



[PS21] [FPSW23, BDPW20, Gen09]

Contributions on mod-uSVP_2



Worst-case to Average-case reduction for id-HSVP

In more details: Number fields and ideals

$\mathbb{Z}^n$	$\mathcal{O}_K = \mathbb{Z}[X]/(X^n+1)$	$\mathbb{Z}[X]/(X^2+1)$
$\boldsymbol{v} = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$	$P(X) = a_0 + a_1 X + \ldots + a_{n-1} X^{n-1}$	$X+2$
$\lVert \boldsymbol{v} \rVert$	$\sqrt{\sum_{i=0}^{n-1} a_i^2}$	$\sqrt{5}$

Definition (Ideal)

A set $\mathfrak{a} \subseteq K$ is an ideal if it is discrete, stable by addition and by multiplication by any element of $\mathcal{O}_K$. **Example:** $(X+2) \cdot \mathcal{O}_K$.

Norm of an ideal: $\mathcal{N}(I) = \mathrm{Vol}(I)/\mathrm{Vol}(\mathcal{O}_K) \in \mathbb{Z}$.

Ideals are lattices!

$$\begin{aligned}\mathcal{L} &= (X+2)\cdot\mathcal{O}_K \\ &= \left\{(X+2)\cdot(a+bX) \bmod X^2+1\right\}\end{aligned}$$

Ideals are lattices!

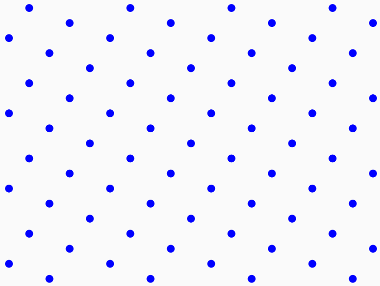
$$
\begin{aligned}
\mathcal{L} &= (X+2)\cdot\mathcal{O}_K \\
&= \left\{(X+2)\cdot(a+bX) \bmod X^2+1\right\} \\
&= \left\{(2a-b)+(a+2b)\cdot X,\ \ a,b\in\mathbb{Z}\right\}
\end{aligned}
$$

Ideals are lattices!

$$
\begin{aligned}
\mathcal{L} &= (X+2)\cdot\mathcal{O}_K \\
&= \left\{(X+2)\cdot(a+bX) \bmod X^2+1\right\} \\
&= \left\{(2a-b)+(a+2b)\cdot X,\ \ a,b\in\mathbb{Z}\right\} \\
&\simeq \left(\begin{smallmatrix} 2 & -1 \\ 1 & 2 \end{smallmatrix}\right)\cdot\mathbb{Z}^2.
\end{aligned}
$$

Ideals are lattices!

$$\begin{aligned}\mathcal{L} &= (X+2)\cdot\mathcal{O}_K \\ &= \left\{(X+2)\cdot(a+bX) \bmod X^2+1\right\} \\ &= \left\{(2a-b)+(a+2b)\cdot X,\ \ a,b\in\mathbb{Z}\right\} \\ &\simeq \left(\begin{smallmatrix}2 & -1\\ 1 & 2\end{smallmatrix}\right)\cdot\mathbb{Z}^2.\end{aligned}$$



The lattice $\mathcal{L}$ associated to $(X+2)\cdot\mathcal{O}_K$.

Ideal Arithmetic: Basic Notions

Let $\mathfrak{a}, \mathfrak{b}$ ideals of K, and $a \in K$.

Principal ideal

$(a) = \{x \cdot a, x \in \mathcal{O}_K\}$.

Multiplication and inverse

$\mathfrak{a} \cdot \mathfrak{b} = \left\{\sum_i a_i \cdot b_i\right\}, \mathfrak{a}^{-1} = \{x \in K, x \cdot \mathfrak{a} \subseteq \mathcal{O}_K\}$. We have that $\mathfrak{a} \cdot \mathfrak{a}^{-1} = \mathcal{O}_K$.

Prime ideals

An ideal $\mathfrak{p} \neq \mathcal{O}_K$ is prime ($\mathfrak{p} \in \mathcal{P}$) if

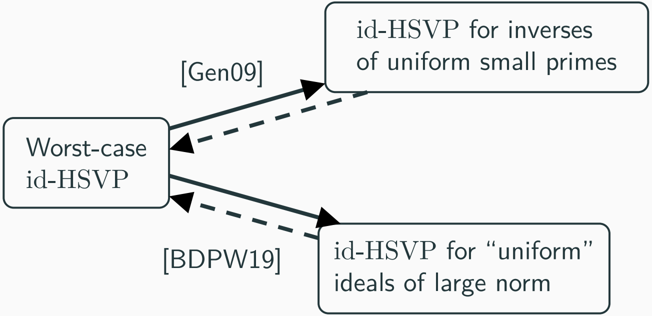
$$\mathfrak{p} = \mathfrak{a} \cdot \mathfrak{b} \Rightarrow \mathfrak{a} = \mathcal{O}_K \text{ or } \mathfrak{b} = \mathcal{O}_K.$$

Is id-HSVP hard for a Random Ideal?

No clear answer. What do you mean by "Random"?

Is id-HSVP hard for a Random Ideal?

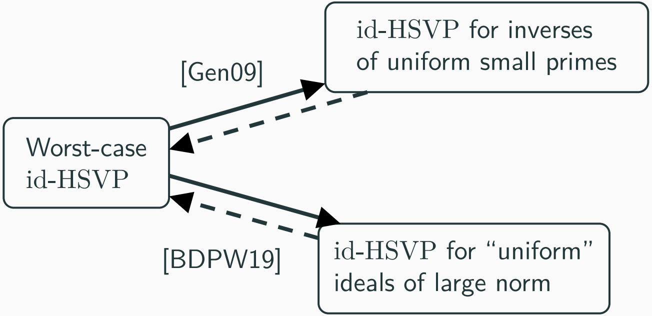
No clear answer. What do you mean by "Random"?



[Gen09] Gentry. A Fully Homomorphic Encryption Scheme. [BDPW20] de Boer, Ducas, Pellet-Mary, Wesolowski. Random self-reducibility of Ideal-SVP via Arakelov random walks.

Is id-HSVP hard for a Random Ideal?

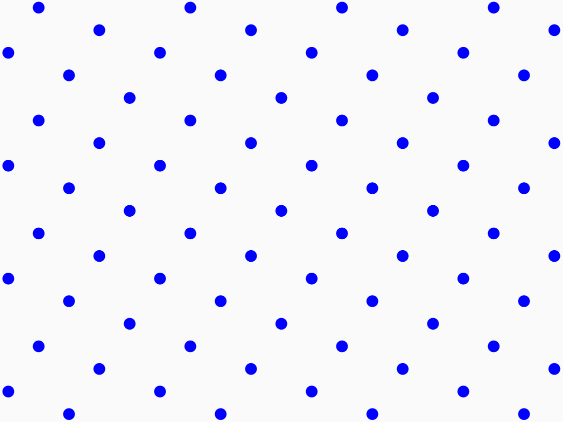
No clear answer. What do you mean by "Random"?



Not natural! We would want the same result for **uniform small prime ideals**.

[Gen09] Gentry. A Fully Homomorphic Encryption Scheme. [BDPW20] de Boer, Ducas, Pellet-Mary, Wesolowski. Random self-reducibility of Ideal-SVP via Arakelov random walks.

How to sample a uniform ideal? [Boe22]

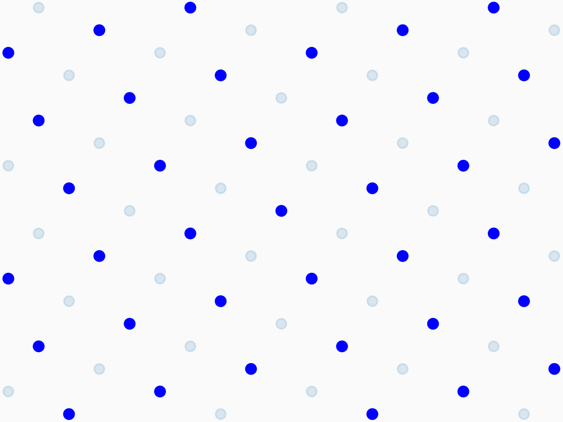


Sampling $\mathfrak{b}$ uniform

Input: any ideal $\mathfrak{a}$.

[Boe22]: K. de Boer. Random Walks on Arakelov Class Groups. PhD thesis, Leiden University, 2022

How to sample a uniform ideal? [Boe22]



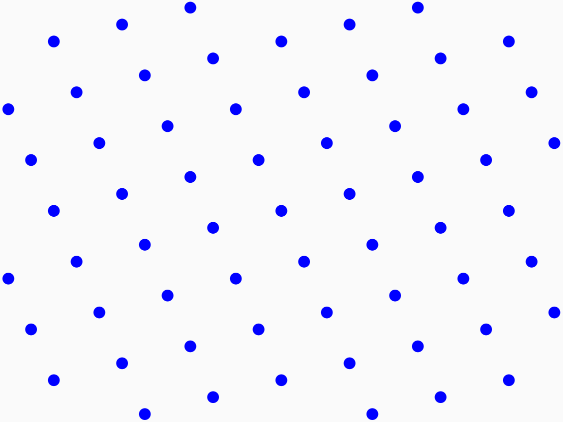
Sampling $\mathfrak{b}$ uniform

Input: any ideal $\mathfrak{a}$.

Sparsification by random $\mathfrak{p}$: $\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$.

[Boe22]: K. de Boer. Random Walks on Arakelov Class Groups. PhD thesis, Leiden University, 2022

How to sample a uniform ideal? [Boe22]



Sampling $\mathfrak{b}$ uniform

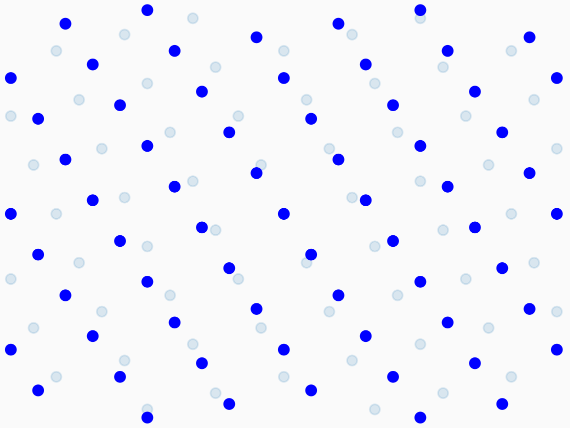
Input: any ideal $\mathfrak{a}$.

Sparsification by random $\mathfrak{p}$: $\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$.

Scaling: $I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$

[Boe22]: K. de Boer. Random Walks on Arakelov Class Groups. PhD thesis, Leiden University, 2022

How to sample a uniform ideal? [Boe22]



Sampling $\mathfrak{b}$ uniform

Input: any ideal $\mathfrak{a}$.

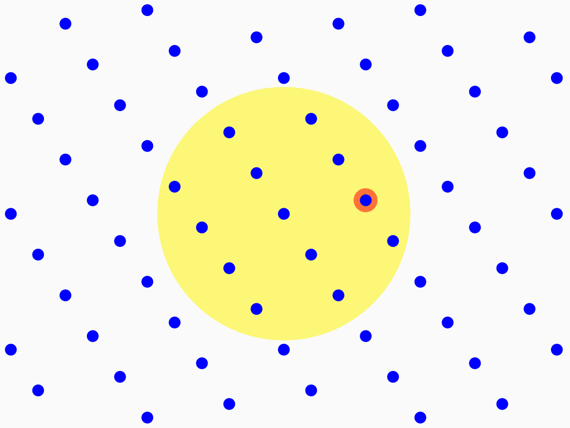
Sparsification by random $\mathfrak{p}$: $\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$.

Scaling: $I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$

Distortion $I_2 = \boldsymbol{D} \cdot I_1$

[Boe22]: K. de Boer. Random Walks on Arakelov Class Groups. PhD thesis, Leiden University, 2022

How to sample a uniform ideal? [Boe22]



Sampling $\mathfrak{b}$ uniform

Input: any ideal $\mathfrak{a}$.

Sparsification by random $\mathfrak{p}$: $\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$.

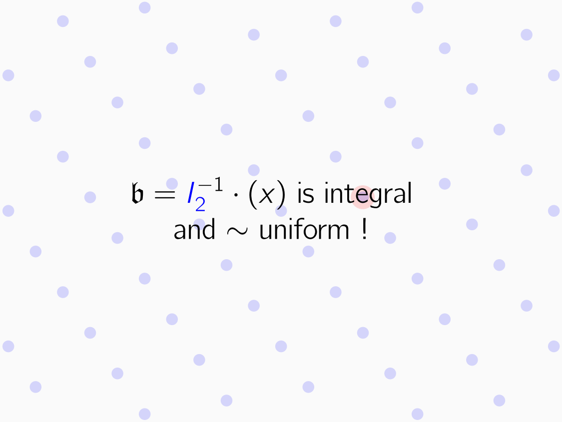
Scaling: $I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$

Distortion $I_2 = \boldsymbol{D} \cdot I_1$

Sample small $x \in I_2$.

[Boe22]: K. de Boer. Random Walks on Arakelov Class Groups. PhD thesis, Leiden University, 2022

How to sample a uniform ideal? [Boe22]



$\mathfrak{b} = I_2^{-1} \cdot (x)$ is integral and $\sim$ uniform !

Sampling $\mathfrak{b}$ uniform

Input: any ideal $\mathfrak{a}$.

Sparsification by random $\mathfrak{p}$: $\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$.

Scaling: $I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$

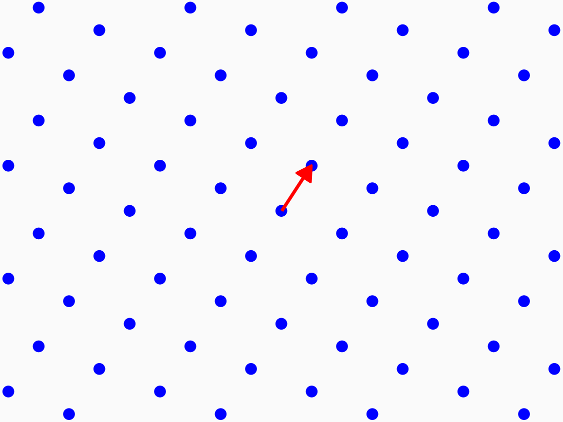
Distortion $I_2 = \boldsymbol{D} \cdot I_1$

Sample small $x \in I_2$**.**

Magic* happens.

[Boe22]: K. de Boer. Random Walks on Arakelov Class Groups. PhD thesis, Leiden University, 2022

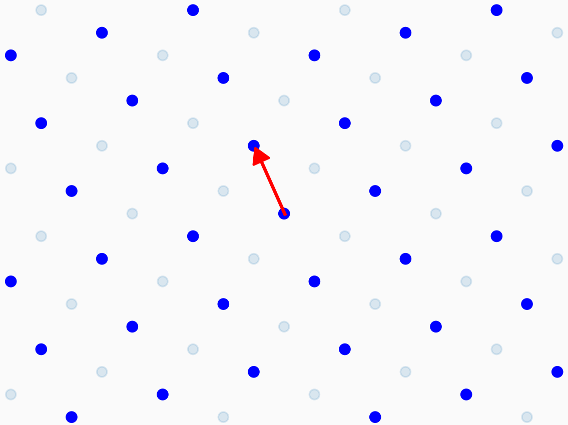
Sampling with a Trapdoor: `SampleIdeal`



Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

Sampling with a Trapdoor: `SampleIdeal`



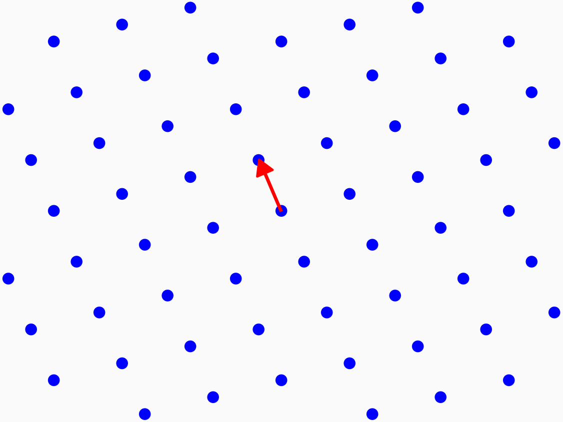
Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

Sparsification by random $\mathfrak{p}$:

$\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$. $\qquad s_{\mathfrak{a}_1} = s_{\mathfrak{a}} \cdot s_{\mathfrak{p}}$.

Sampling with a Trapdoor: `SampleIdeal`



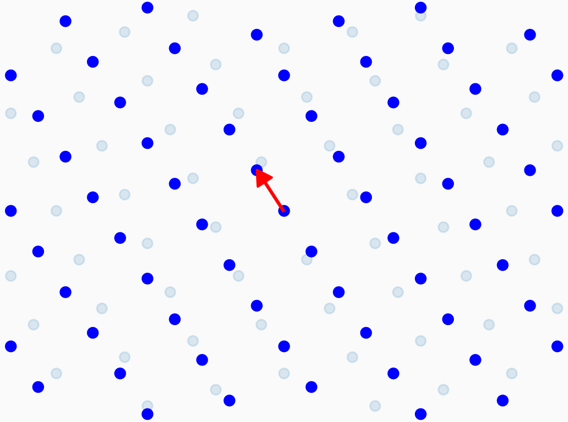
Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

Sparsification by random $\mathfrak{p}$:
$\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$. $\qquad s_{\mathfrak{a}_1} = s_{\mathfrak{a}} \cdot s_{\mathfrak{p}}$.

Scaling:
$I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$ $\qquad s_{I_1} = s_{\mathfrak{a}_1} / (\cdots)$.

Sampling with a Trapdoor: `SampleIdeal`



Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

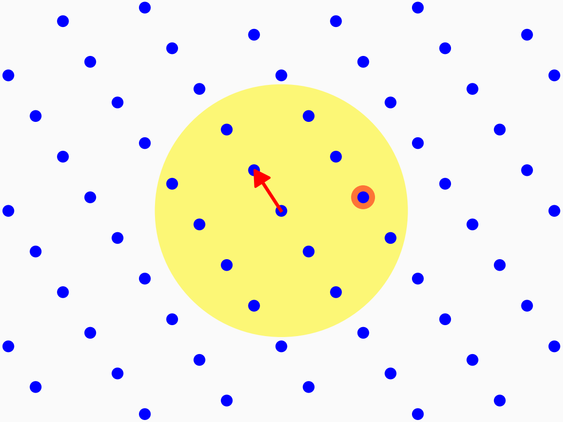
Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

Sparsification by random $\mathfrak{p}$:
$\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$. $\qquad s_{\mathfrak{a}_1} = s_{\mathfrak{a}} \cdot s_{\mathfrak{p}}$.

Scaling:
$I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$ $\qquad s_{I_1} = s_{\mathfrak{a}_1} / (\cdots)$.

Distortion
$I_2 = \boldsymbol{D} \cdot I_1$ $\qquad s_{I_2} = \boldsymbol{D} \cdot s_{I_1}$.

Sampling with a Trapdoor: `SampleIdeal`



Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

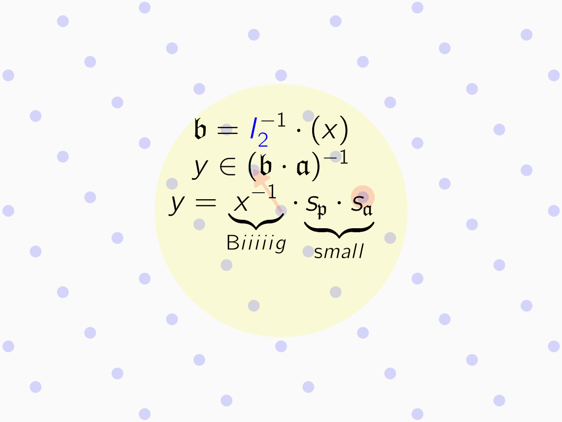
Sparsification by random $\mathfrak{p}$:
$\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$. $\qquad s_{\mathfrak{a}_1} = s_{\mathfrak{a}} \cdot s_{\mathfrak{p}}$.

Scaling:
$I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$ $\qquad s_{I_1} = s_{\mathfrak{a}_1}/(\cdots)$.

Distortion
$I_2 = \boldsymbol{D} \cdot I_1$ $\qquad s_{I_2} = \boldsymbol{D} \cdot s_{I_1}$.

Sample small $x \in I_2$. $\qquad y = x^{-1} \cdot s_{\mathfrak{p}} \cdot s_{\mathfrak{a}}$

Sampling with a Trapdoor: `SampleIdeal`



Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

Sparsification by random $\mathfrak{p}$:
$\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$. $\qquad s_{\mathfrak{a}_1} = s_{\mathfrak{a}} \cdot s_{\mathfrak{p}}$.

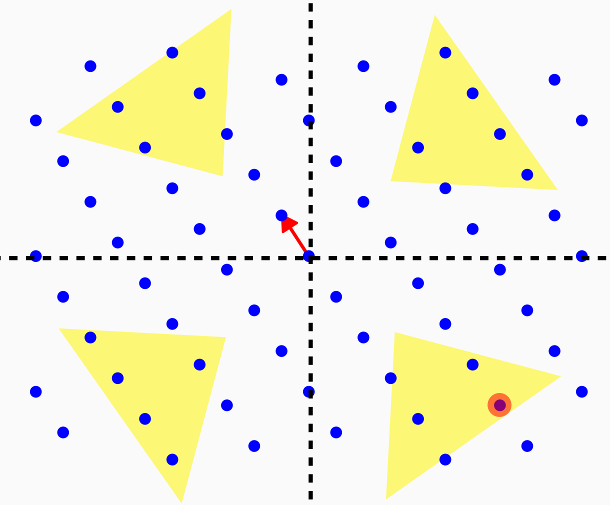
Scaling:
$I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$ $\qquad s_{I_1} = s_{\mathfrak{a}_1} / (\cdots)$.

Distortion
$I_2 = \boldsymbol{D} \cdot I_1$ $\qquad s_{I_2} = \boldsymbol{D} \cdot s_{I_1}$.

Sample small $x \in I_2$. $\qquad y = x^{-1} \cdot s_{\mathfrak{p}} \cdot s_{\mathfrak{a}}$

Magic happens?

Sampling with a Trapdoor: `SampleIdeal`



Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

Sparsification by random $\mathfrak{p}$:
$\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$. $\qquad s_{\mathfrak{a}_1} = s_{\mathfrak{a}} \cdot s_{\mathfrak{p}}$.

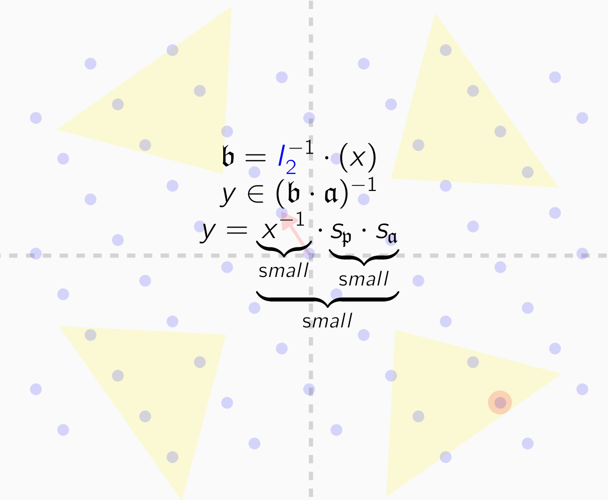
Scaling:
$I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$ $\qquad s_{I_1} = s_{\mathfrak{a}_1} / (\cdots)$.

Distortion
$I_2 = \boldsymbol{D} \cdot I_1$ $\qquad s_{I_2} = \boldsymbol{D} \cdot s_{I_1}$.

Sample $x \in I \bigcap \mathcal{B}$. $\qquad y = x^{-1} \cdot s_{\mathfrak{p}} \cdot s_{\mathfrak{a}}$

Magic happens?

Sampling with a Trapdoor: $\texttt{SampleIdeal}_{\mathcal{B}}$



$$\mathfrak{b} = I_2^{-1} \cdot (x)$$

$$y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$$

$$y = \underbrace{\underbrace{x^{-1}}_{small} \cdot \underbrace{s_{\mathfrak{p}} \cdot s_{\mathfrak{a}}}_{small}}_{small}$$

Sampling $(\mathfrak{b}, y)$ with $y \in (\mathfrak{b} \cdot \mathfrak{a})^{-1}$ small

Input: any ideal $\mathfrak{a}$ $\qquad s_{\mathfrak{a}} \in \mathfrak{a}$.

Sparsification by random $\mathfrak{p}$:
$\mathfrak{a}_1 = \mathfrak{a} \cdot \mathfrak{p}$. $\qquad s_{\mathfrak{a}_1} = s_{\mathfrak{a}} \cdot s_{\mathfrak{p}}$.

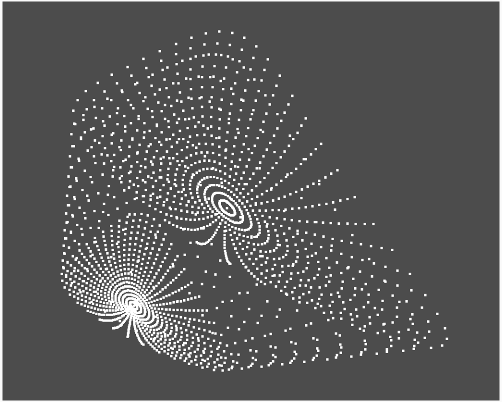
Scaling:
$I_1 = \mathfrak{a}_1 / \mathcal{N}(\mathfrak{a}_1)^{1/d}$ $\qquad s_{I_1} = s_{\mathfrak{a}_1}/(\cdots)$.

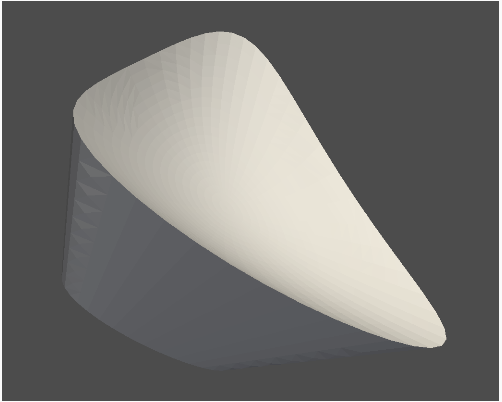
Distortion
$I_2 = \boldsymbol{D} \cdot I_1$ $\qquad s_{I_2} = \boldsymbol{D} \cdot s_{I_1}$.

Sample $x \in I \bigcap \mathcal{B}$. $\qquad y = x^{-1} \cdot s_{\mathfrak{p}} \cdot s_{\mathfrak{a}}$

Magic happens

The shape $\mathcal{B}$





$$\mathcal{B}^{\eta}_{A,B} = \left\{ x \in K_{\mathbb{R}}, \quad |\mathcal{N}(x)| \in [A, B], \quad \left\| \mathrm{Ln}\left(\frac{x}{\mathcal{N}(x)^{1/d}} \right) \right\|_2 \leq \log(\eta) \right\}$$

Taking a step back: $\texttt{SampleIdeal}_{\mathcal{B}}$

$\texttt{SampleIdeal}_{\mathcal{B}}$

1. Takes as input $\mathfrak{a} \subseteq \mathcal{O}_K$ and $s_{\mathfrak{a}} \in \mathfrak{a}$ small.
2. Output $\mathfrak{b} \subseteq \mathcal{O}_K$ uniform and $y \in \mathfrak{b}^{-1} \cdot \mathfrak{a}^{-1}$ small.

Taking a step back: $\texttt{SampleIdeal}_{\mathcal{B}}$

$\texttt{SampleIdeal}_{\mathcal{B}}$

1. Takes as input $\mathfrak{a} \subseteq \mathcal{O}_K$ and $s_{\mathfrak{a}} \in \mathfrak{a}$ small.
2. Output $\mathfrak{b} \subseteq \mathcal{O}_K$ uniform and $y \in \mathfrak{b}^{-1} \cdot \mathfrak{a}^{-1}$ small.

Now if we can find $s_{\mathfrak{b}} \in \mathfrak{b}$ small, then $s_{\mathfrak{b}} \cdot y$ is small and

$$s_{\mathfrak{b}} \cdot y \in \mathfrak{b} \cdot \mathfrak{b}^{-1} \cdot \mathfrak{a}^{-1} = \mathfrak{a}^{-1}$$

Taking a step back: $\texttt{SampleIdeal}_{\mathcal{B}}$

$\texttt{SampleIdeal}_{\mathcal{B}}$

1. Takes as input $\mathfrak{a} \subseteq \mathcal{O}_K$ and $s_{\mathfrak{a}} \in \mathfrak{a}$ small.
2. Output $\mathfrak{b} \subseteq \mathcal{O}_K$ uniform and $y \in \mathfrak{b}^{-1} \cdot \mathfrak{a}^{-1}$ small.

Now if we can find $s_{\mathfrak{b}} \in \mathfrak{b}$ small, then $s_{\mathfrak{b}} \cdot y$ is small and

$$s_{\mathfrak{b}} \cdot y \in \mathfrak{b} \cdot \mathfrak{b}^{-1} \cdot \mathfrak{a}^{-1} = \mathfrak{a}^{-1}$$

$$\boxed{\text{id-HSVP}(\mathfrak{a}^{-1}) \xrightarrow{\texttt{SampleIdeal}_{\mathcal{B}}} \text{id-HSVP}(\mathfrak{a}) + \text{id-HSVP}(\mathfrak{b})}$$

$$\begin{array}{c} \boxed{\text{Worst-case id-HSVP}} \\ \downarrow \text{[Gen09]} \\ \boxed{\text{id-HSVP on uniform } \mathfrak{p}^{-1}} \\ \downarrow \texttt{SampleIdeal}_{\mathcal{B}} \\ \boxed{\text{id-HSVP on uniform } \mathfrak{p}} \end{array}$$

The $\mathcal{P}^{-1}$-ideal-SVP to $\mathcal{P}$-ideal-SVP reduction

The oracle $\mathcal{O}$ solves id-HSVP for $\mathfrak{p}$ uniform prime of norm in $[A, B]$.

Input: An ideal $I = \mathfrak{p}^{-1}$ with $\mathfrak{p}$ uniform prime of norm in $[A, B]$.
Output: $x \in \mathfrak{p}^{-1} \setminus \{0\}$ small.

Let $s_{\mathfrak{p}} = \mathcal{O}(\mathfrak{p})$.
Let $(\mathfrak{b}, y) = \mathtt{SampleIdeal}_{\mathcal{B}}(\mathfrak{p}, s_{\mathfrak{p}})$. $\triangleright \|y\|$ *small*
if $\mathfrak{b}$ is not prime **then**
  Fail.
Let $s_{\mathfrak{b}} = \mathcal{O}(\mathfrak{b})$. $\triangleright \|s_{\mathfrak{b}}\|$ *small*
Return $\underbrace{s_{\mathfrak{b}}}_{\in \mathfrak{b}} \cdot \underbrace{y}_{\in (\mathfrak{b}\cdot\mathfrak{p})^{-1}} \in \mathfrak{p}^{-1}$. $\triangleright \|y \cdot s_{\mathfrak{b}}\|$ *small*

The $\mathcal{P}^{-1}$-ideal-SVP to $\mathcal{P}$-ideal-SVP reduction

The oracle $\mathcal{O}$ solves id-HSVP for $\mathfrak{p}$ uniform prime of norm in $[A, B]$.

Input: An ideal $I = \mathfrak{p}^{-1}$ with $\mathfrak{p}$ uniform prime of norm in $[A, B]$.
Output: $x \in \mathfrak{p}^{-1} \setminus \{0\}$ small.

1: Let $s_{\mathfrak{p}} = \mathcal{O}(\mathfrak{p})$.
2: Let $(\mathfrak{b}, y) = \mathtt{SampleIdeal}_{\mathcal{B}}(\mathfrak{p}, s_{\mathfrak{p}})$. ▷ $\|y\|$ *small*
3: **if** $\mathfrak{b}$ is not prime **then**
4: Fail.
5: Let $s_{\mathfrak{b}} = \mathcal{O}(\mathfrak{b})$. ▷ $\|s_{\mathfrak{b}}\|$ *small*
6: **Return** $\underbrace{s_{\mathfrak{b}}}_{\in \mathfrak{b}} \cdot \underbrace{y}_{\in (\mathfrak{b} \cdot \mathfrak{p})^{-1}} \in \mathfrak{p}^{-1}$. ▷ $\|y \cdot s_{\mathfrak{b}}\|$ *small*

The $\mathcal{P}^{-1}$-ideal-SVP to $\mathcal{P}$-ideal-SVP reduction

The oracle $\mathcal{O}$ solves id-HSVP for $\mathfrak{p}$ uniform prime of norm in $[A, B]$.

Input: An ideal $I = \mathfrak{p}^{-1}$ with $\mathfrak{p}$ uniform prime of norm in $[A, B]$.
Output: $x \in \mathfrak{p}^{-1} \setminus \{0\}$ small.

1: Let $s_{\mathfrak{p}} = \mathcal{O}(\mathfrak{p})$.
2: Let $(\mathfrak{b}, y) = \texttt{SampleIdeal}_{\mathcal{B}}(\mathfrak{p}, s_{\mathfrak{p}})$. ▷ $\|y\|$ *small*
3: **if** $\mathfrak{b}$ is not prime **then**
4: Fail.
5: Let $s_{\mathfrak{b}} = \mathcal{O}(\mathfrak{b})$. ▷ $\|s_{\mathfrak{b}}\|$ *small*
6: **Return** $\underbrace{s_{\mathfrak{b}}}_{\in \mathfrak{b}} \cdot \underbrace{y}_{\in (\mathfrak{b}\cdot\mathfrak{p})^{-1}} \in \mathfrak{p}^{-1}$. ▷ $\|y \cdot s_{\mathfrak{b}}\|$ *small*

The $\mathcal{P}^{-1}$-ideal-SVP to $\mathcal{P}$-ideal-SVP reduction

The oracle $\mathcal{O}$ solves id-HSVP for $\mathfrak{p}$ uniform prime of norm in $[A, B]$.

Input: An ideal $I = \mathfrak{p}^{-1}$ with $\mathfrak{p}$ uniform prime of norm in $[A, B]$.
Output: $x \in \mathfrak{p}^{-1} \setminus \{0\}$ small.

1: Let $s_\mathfrak{p} = \mathcal{O}(\mathfrak{p})$.
2: Let $(\mathfrak{b}, y) = \mathtt{SampleIdeal}_\mathcal{B}(\mathfrak{p}, s_\mathfrak{p})$. ▷ $\|y\|$ *small*
3: **if** $\mathfrak{b}$ is not prime **then**
4: Fail.
5: Let $s_\mathfrak{b} = \mathcal{O}(\mathfrak{b})$. ▷ $\|s_\mathfrak{b}\|$ *small*
6: **Return** $\underbrace{s_\mathfrak{b}}_{\in \mathfrak{b}} \cdot \underbrace{y}_{\in (\mathfrak{b}\cdot\mathfrak{p})^{-1}} \in \mathfrak{p}^{-1}$. ▷ $\|y \cdot s_\mathfrak{b}\|$ *small*

The $\mathcal{P}^{-1}$-ideal-SVP to $\mathcal{P}$-ideal-SVP reduction

The oracle $\mathcal{O}$ solves id-HSVP for $\mathfrak{p}$ uniform prime of norm in $[A, B]$.

Input: An ideal $I = \mathfrak{p}^{-1}$ with $\mathfrak{p}$ uniform prime of norm in $[A, B]$.
Output: $x \in \mathfrak{p}^{-1} \setminus \{0\}$ small.

1: Let $s_{\mathfrak{p}} = \mathcal{O}(\mathfrak{p})$.
2: Let $(\mathfrak{b}, y) = \texttt{SampleIdeal}_{\mathcal{B}}(\mathfrak{p}, s_{\mathfrak{p}})$. ▷ $\|y\|$ *small*
3: **if** $\mathfrak{b}$ is not prime **then**
4: Fail.
5: Let $s_{\mathfrak{b}} = \mathcal{O}(\mathfrak{b})$. ▷ $\|s_{\mathfrak{b}}\|$ *small*
6: **Return** $\underbrace{s_{\mathfrak{b}}}_{\in \mathfrak{b}} \cdot \underbrace{y}_{\in (\mathfrak{b} \cdot \mathfrak{p})^{-1}} \in \mathfrak{p}^{-1}$. ▷ $\|y \cdot s_{\mathfrak{b}}\|$ *small*

The $\mathcal{P}^{-1}$-ideal-SVP to $\mathcal{P}$-ideal-SVP reduction

The oracle $\mathcal{O}$ solves id-HSVP for $\mathfrak{p}$ uniform prime of norm in $[A, B]$.

Input: An ideal $I = \mathfrak{p}^{-1}$ with $\mathfrak{p}$ uniform prime of norm in $[A, B]$.
Output: $x \in \mathfrak{p}^{-1} \setminus \{0\}$ small.

Let $s_{\mathfrak{p}} = \mathcal{O}(\mathfrak{p})$.
Let $(\mathfrak{b}, y) = \texttt{SampleIdeal}_{\mathcal{B}}(\mathfrak{p}, s_{\mathfrak{p}})$. $\triangleright \|y\|$ *small*
if $\mathfrak{b}$ is not prime **then**
    Fail.
Let $s_{\mathfrak{b}} = \mathcal{O}(\mathfrak{b})$. $\triangleright \|s_{\mathfrak{b}}\|$ *small*
Return $\underbrace{s_{\mathfrak{b}}}_{\in \mathfrak{b}} \cdot \underbrace{y}_{\in (\mathfrak{b}\cdot\mathfrak{p})^{-1}} \in \mathfrak{p}^{-1}$. $\triangleright \|y \cdot s_{\mathfrak{b}}\|$ *small*

Contributions and open problems for $\mathrm{id}\text{-}\mathrm{HSVP}$

Contributions:

- New ideal sampling algorithm.
- Solving $\mathrm{id}\text{-}\mathrm{HSVP}$ on average over primes $\simeq$ solving $\mathrm{id}\text{-}\mathrm{HSVP}$ for any ideal.

Contributions and open problems for $\mathrm{id\text{-}HSVP}$

Contributions:

- New ideal sampling algorithm.
- Solving $\mathrm{id\text{-}HSVP}$ on average over primes $\simeq$ solving $\mathrm{id\text{-}HSVP}$ for any ideal.

Open problems:

- Can we have such reduction without factoring?
- Can we get rid of the cost dependency in ρ_K?

Conclusion and Perspectives

Taking a step back

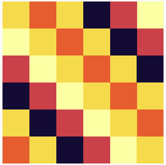
- Structured lattice problems $\rightarrow$ better performance for cryptography.
- **But** might introduce weaknesses.
- We worked on ranks 1 and 2.

Taking a step back

- Structured lattice problems → better performance for cryptography.
- **But** might introduce weaknesses.
- We worked on ranks 1 and 2.

Rank 1: id-HSVP

- Proposed a new sampling algorithm.
- Proved that a "natural" distribution is secure.



Taking a step back

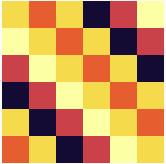
- Structured lattice problems $\rightarrow$ better performance for cryptography.
- **But** might introduce weaknesses.
- We worked on ranks 1 and 2.

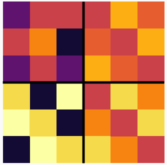
Rank 1: $\mathrm{id\text{-}HSVP}$

- Proposed a new sampling algorithm.
- Proved that a “natural” distribution is secure.

Rank 2: $\mathrm{mod\text{-}uSVP}_2$

- Proposed a “natural” distribution of instances.
- Proved a worst-case to average-case reduction for this distribution.





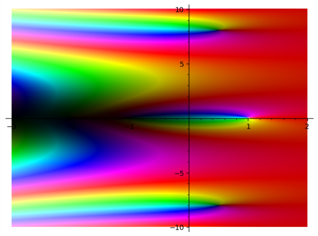
Other contributions of this thesis

Reduction between
mod-uSVP_2 and NTRU.

Other contributions of this thesis

Reduction between
$\mathrm{mod\text{-}uSVP}_2$ and NTRU.

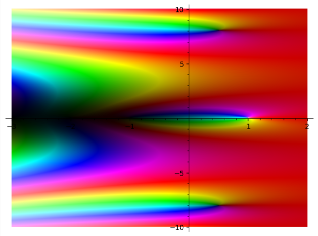
A new bound on
ideal-counting function.



Other contributions of this thesis

Reduction between
$\mathrm{mod\text{-}uSVP}_2$ and NTRU.

A new bound on
ideal-counting function.



A more generic average-case
reduction for $\mathrm{id\text{-}HSVP}$.

$$\boxed{\mathcal{U}(\mathcal{W}^{-1})\text{-id-HSVP}}$$

$$\downarrow$$

$$\boxed{\mathcal{U}(\mathcal{W})\text{-id-HSVP}}$$

$$+$$

$$\boxed{\mathcal{U}(\mathcal{I}_{A,B})\text{-id-HSVP}}$$

Perspectives

Reductions

- Understand gap between rank 1 and 2 (γ-mod-uSVP$_2$?).
- Go to higher rank: mod-NTRU$_{n,m}$, mod-uSVP$_{n,m}$.
- Other structured problems e.g., mod-LIP.

Perspectives

Reductions

- Understand gap between rank 1 and 2 (γ-mod-uSVP$_2$?).
- Go to higher rank: mod-NTRU$_{n,m}$, mod-uSVP$_{n,m}$.
- Other structured problems e.g., mod-LIP.

Links to Number Theory

- Sampling prime ideals without factoring.
- Haar distributions on compact sets of modules.
- Are some fields easier? (e.g. $\zeta_K(2)$ or Δ_K small...).
- Improve the error bound on $N_K(\cdot)$.

Perspectives

Reductions

- Understand gap between rank 1 and 2 (γ-mod-uSVP$_2$?).
- Go to higher rank: mod-NTRU$_{n,m}$, mod-uSVP$_{n,m}$.
- Other structured problems e.g., mod-LIP.

Links to Number Theory

- Sampling prime ideals without factoring.
- Haar distributions on compact sets of modules.
- Are some fields easier? (e.g. $\zeta_K(2)$ or Δ_K small...).
- Improve the error bound on $N_K(\cdot)$.

Other directions

- Cryptanalysis of "with hint" assumptions.
- Real-world: assumptions used in socially beneficial cryptography (e.g. anamorphic encryption).
- Look for weaknesses in PQ crypto implementations.

Perspectives

Reductions

- Understand gap between rank 1 and 2 (γ-mod-uSVP$_2$?).
- Go to higher rank: mod-NTRU$_{n,m}$, mod-uSVP$_{n,m}$.
- Other structured problems e.g., mod-LIP.

Links to Number Theory

- Sampling prime ideals without factoring.
- Haar distributions on compact sets of modules.
- Are some fields easier? (e.g. $\zeta_K(2)$ or Δ_K small...).
- Improve the error bound on $N_K(\cdot)$.

Other directions

- Cryptanalysis of "with hint" assumptions.
- Real-world: assumptions used in socially beneficial cryptography (e.g. anamorphic encryption).
- Look for weaknesses in PQ crypto implementations.

Thank you for your attention. I would be happy to answer your questions.

Pour ma famille : c'est un bon moment pour fuir.

References

References i

M. R. Albrecht and A. Deo.
Large Modulus Ring-LWE $\geq$ Module-LWE.
In *ASIACRYPT*, 2017.

K. de Boer, L. Ducas, A. Pellet-Mary, and B. Wesolowski.
Random self-reducibility of Ideal-SVP via Arakelov random walks.
In *CRYPTO*, 2020.

K. de Boer.
Random Walks on Arakelov Class Groups.
PhD thesis, Leiden University, 2022.
Available on request from the author.

J. Felderhoff, A. Pellet-Mary, and D. Stehlé.
On module unique-SVP and NTRU.
In *ASIACRYPT*, 2022.

References ii

J. Felderhoff, A. Pellet-Mary , D. Stehlé, and B. Wesolowski.
Ideal-SVP is hard for small-norm uniform prime ideals.
In *TCC*, 2023.

C. Gentry.
A Fully Homomorphic Encryption Scheme.
PhD thesis, Stanford University, 2009.

J. Hoffstein, J. Pipher, and J. H. Silverman.
NTRU: a ring based public key cryptosystem.
In *ANTS*, 1998.

A. Langlois and D. Stehlé.
Worst-case to average-case reductions for module lattices.
Design Code and Cryptography, 2015.

References iii

A. Pellet-Mary and D. Stehlé.
On the hardness of the NTRU problem.
In *ASIACRYPT*, 2021.

Extra Frames

Rounding Module in $K_{\mathbb{R}}$

The "good basis" is randomized, but not the "bad" one.

	Basis	Short vector
	$\begin{pmatrix} \widetilde{b}_{11} & \widetilde{b}_{12} \\ \widetilde{b}_{21} & \widetilde{b}_{22} \end{pmatrix} \in {K_{\mathbb{R}}}^{2\times 2}$	$\widetilde{\boldsymbol{s}} = \begin{bmatrix} \widetilde{u} \\ \widetilde{v} \end{bmatrix}$
$(M^\vee)^2 \ni$	$(\lambda \boldsymbol{I} + \boldsymbol{\varepsilon}) \times \downarrow$	$(\lambda \boldsymbol{I} + \boldsymbol{\varepsilon}) \times \downarrow$
	$\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \in {\mathcal{O}_K}^{2\times 2}$	$\boldsymbol{s} = (\lambda \boldsymbol{I} + \boldsymbol{\varepsilon})\widetilde{\boldsymbol{s}} \in {\mathcal{O}_K}^2$

Lemma (definition of the dual)

If $\boldsymbol{u}, \boldsymbol{v} \in M^\vee$, then $[\boldsymbol{u}, \boldsymbol{v}]^T \cdot M \subset {\mathcal{O}_K}^2$.

Then take HNF.

NTRU

We work with elements of $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$ for $n = 2^r$.

Definition (NTRU_q)

Let $f, g \in \mathcal{O}_K$ with coefficients $\ll \sqrt{q}$ and f invertible mod q.
Given $h \in \mathcal{O}_K$ such that $f \cdot h = g$ mod q, find a small multiple of (f, g).

Proposed first in [HPS98].
Used in NIST's post-quantum standardization process: **NTRU** and **NTRUPrime**.

Advantages:

- Small keys.
- Fast encryption/decryption (much faster than RSA).
- Old.

The NTRU module

Given $h \in \mathcal{O}_K$, the set of solutions for (f, g) is

$$M = \left\{ (f_0, g_0)^T \in {\mathcal{O}_K}^2, \quad f_0 \cdot h = g_0 \bmod q \right\}$$

This is a module generated by the matrix

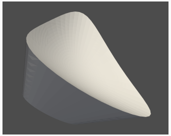
$$\boldsymbol{B} = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix}$$

Solving NTRU is finding a short non-zero vector in M.

Big gap: NTRU is an instance of mod-uSVP_2

$$\lambda_1(M) \leq \|(f, g)^T\| \ll \sqrt{q} \quad \text{versus} \quad \lambda_2(M) \geq \det(\boldsymbol{B})/\lambda_1 \gg \sqrt{q}.$$

What does “well chosen” mean?



1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements must be balanced.

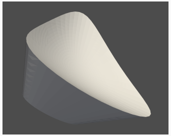
Balanced elements (for Minkowski embedding)

$x \in K$ is balanced if for all i,

$$\frac{1}{\eta} \leq \frac{x_i}{\prod_j x_j^{1/d}} \leq \eta.$$

This is the same as $x \approx \mathcal{N}(x)^{1/d} \cdot (1, \dots, 1)$.

What does “well chosen” mean?



1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements must be balanced.

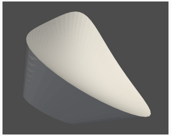
Balanced elements (for Minkowski embedding)

$x \in K$ is balanced if for all i,

$$\frac{1}{\eta} \leq \frac{x_i}{\prod_j x_j^{1/d}} \leq \eta.$$

This is the same as $x \approx \mathcal{N}(x)^{1/d} \cdot (1, \ldots, 1)$.

What does "well chosen" mean?



1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements must be balanced.

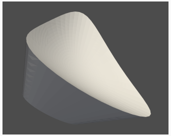
Balanced elements (for Minkowski embedding)

$x \in K$ is balanced if for all i,

$$\frac{1}{\eta} \leq \frac{x_i}{\prod_j x_j^{1/d}} \leq \eta.$$

This is the same as $x \approx \mathcal{N}(x)^{1/d} \cdot (1, \ldots, 1)$.

What does "well chosen" mean?



1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements must be balanced.

Balanced elements (for Minkowski embedding)

$x \in K$ is balanced if for all i,

$$\frac{1}{\eta} \leq \frac{x_i}{\prod_j x_j^{1/d}} \leq \eta.$$

This is the same as $x \approx \mathcal{N}(x)^{1/d} \cdot (1, \ldots, 1)$.

In [BDPW20]: $\mathcal{B}_\infty(r)$: verifies items 1 and 2 but not 3!

Our shape

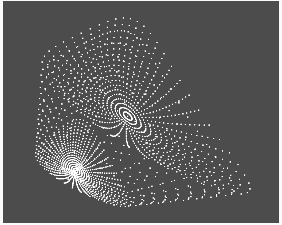
Reminder: conditions for being **well chosen**:

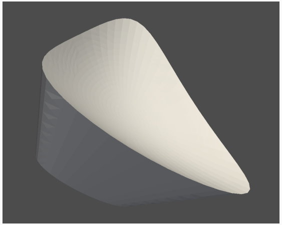
1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements are balanced.

Our shape

Reminder: conditions for being **well chosen**:

1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements are balanced.



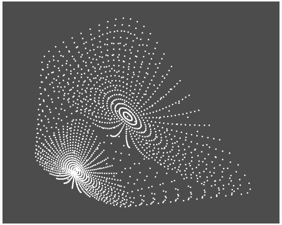


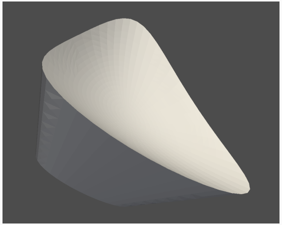
$$\mathcal{B}^{\eta}_{A,B} = \left\{ x \in K_{\mathbb{R}}, \quad |\mathcal{N}(x)| \in [A, B], \quad \left\| \mathrm{Ln}\left(\frac{x}{\mathcal{N}(x)^{1/d}} \right) \right\|_2 \leq \log(\eta) \right\}$$

Our shape

Reminder: conditions for being **well chosen**:

1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements are balanced.



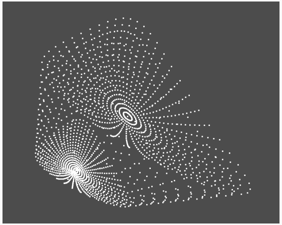


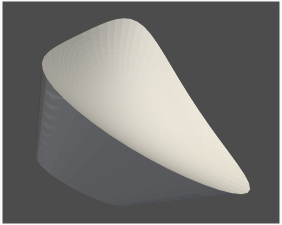
$$\mathcal{B}^{\eta}_{A,B} = \left\{ x \in K_{\mathbb{R}}, \quad |\mathcal{N}(x)| \in [A, B], \quad \left\| \mathrm{Ln}\left(\frac{x}{\mathcal{N}(x)^{1/d}} \right) \right\|_2 \leq \log(\eta) \right\}$$

Our shape

Reminder: conditions for being **well chosen**:

1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements are balanced.



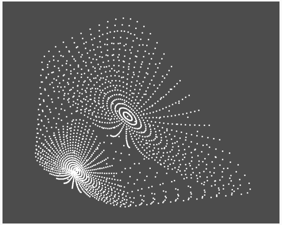


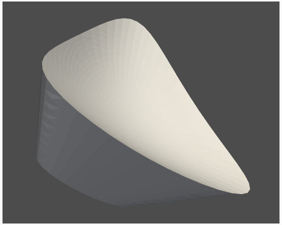
$$\mathcal{B}^{\eta}_{A,B} = \left\{ x \in K_{\mathbb{R}}, \quad |\mathcal{N}(x)| \in [A, B], \quad \left\| \mathrm{Ln}\left(\frac{x}{\mathcal{N}(x)^{1/d}} \right) \right\|_2 \leq \log(\eta) \right\}$$

Our shape

Reminder: conditions for being **well chosen**:

1. $|\mathcal{B}_{A,B} \bigcap \mathfrak{a}|$ does not depend on $\mathfrak{a}$ (too much).
2. $\mathrm{Vol}(\mathrm{Ln}(\mathcal{B}_{A,B}) \bigcap \{\sum x_i = t\})$ is constant for $t \in [A, B]$.
3. Its elements are balanced.





$$\mathcal{B}^{\eta}_{A,B} = \left\{ x \in K_{\mathbb{R}}, \quad |\mathcal{N}(x)| \in [A, B], \quad \left\| \mathrm{Ln}\left(\frac{x}{\mathcal{N}(x)^{1/d}} \right) \right\|_2 \leq \log(\eta) \right\}$$