Lattice fundamentals

Fernando Virdia — `https://fundamental.domains`

EPFL-ETH Summer School on Lattice-based Cryptography, July 2025

Housekeeping

Interaction

Please feel free to interrupt me and ask questions.

Housekeeping

Interaction

Please feel free to interrupt me and ask questions. When I first saw this stuff, I didn't get it either.

Housekeeping

Interaction

Please feel free to interrupt me and ask questions. When I first saw this stuff, I didn't get it either.

Slides

I'll upload them to my personal website, `https://fundamental.domains`

Housekeeping

Interaction

Please feel free to interrupt me and ask questions. When I first saw this stuff, I didn't get it either.

Slides

I'll upload them to my personal website, `https://fundamental.domains`

Sources

These notes have been adapted from pre-existing material, mainly [1], [2].
References at the end of the deck.

Motivation

- Lattices have a long history in mathematics and physics.

Motivation

- Lattices have a long history in mathematics and physics.
- In cryptography, we've been using lattice-related problems and techniques for decades.

Motivation

- Lattices have a long history in mathematics and physics.
- In cryptography, we've been using lattice-related problems and techniques for decades.
  - Subset-sum and hidden-number problems.

Motivation

- Lattices have a long history in mathematics and physics.
- In cryptography, we've been using lattice-related problems and techniques for decades.
  - Subset-sum and hidden-number problems.
  - Partial key-exposure attacks on RSA and ECDSA.

Motivation

- Lattices have a long history in mathematics and physics.
- In cryptography, we've been using lattice-related problems and techniques for decades.
  - Subset-sum and hidden-number problems.
  - Partial key-exposure attacks on RSA and ECDSA.
  - ePrint 1996/9: Collision-Free Hashing from Lattice Problems.
  - ePrint 1996/16: Public-Key Cryptosystems from Lattice Reduction Problems.
  - ePrint 1997/8: Factoring via Strong Lattice Reduction Algorithms.

Motivation

- Lattices have a long history in mathematics and physics.
- In cryptography, we've been using lattice-related problems and techniques for decades.
  - Subset-sum and hidden-number problems.
  - Partial key-exposure attacks on RSA and ECDSA.
  - ePrint 1996/9: Collision-Free Hashing from Lattice Problems.
  - ePrint 1996/16: Public-Key Cryptosystems from Lattice Reduction Problems.
  - ePrint 1997/8: Factoring via Strong Lattice Reduction Algorithms.
- Today, an extremely popular subject.
- They are used to build PQC, FHE, iO, proof systems...

Objective

Today I'll try to introduce you to the basics of lattices in cryptography.

Objective

Today I'll try to introduce you to the basics of lattices in cryptography. We'll cover

1. Mathematical definitions and properties.

Objective

Today I'll try to introduce you to the basics of lattices in cryptography. We'll cover

1. Mathematical definitions and properties.
2. Computational problems and related hardness assumptions.

Objective

Today I'll try to introduce you to the basics of lattices in cryptography. We'll cover

1. Mathematical definitions and properties.
2. Computational problems and related hardness assumptions.
3. 10.000m view of how to attack those assumptions.

Objective

Today I'll try to introduce you to the basics of lattices in cryptography. We'll cover

1. Mathematical definitions and properties.
2. Computational problems and related hardness assumptions.
3. 10.000m view of how to attack those assumptions.
4. If time permits: a small cryptanalysis lab.

Objective

Today I'll try to introduce you to the basics of lattices in cryptography. We'll cover

1. Mathematical definitions and properties.
2. Computational problems and related hardness assumptions.
3. 10.000m view of how to attack those assumptions.
4. If time permits: a small cryptanalysis lab.

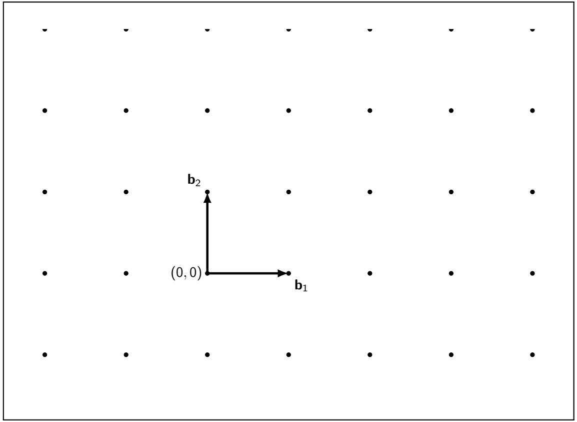
Enough said, let's start.

Definitions and basic properties

Fernando Virdia — `https://fundamental.domains`
EPFL-ETH Summer School on Lattice-based Cryptography, July 2025

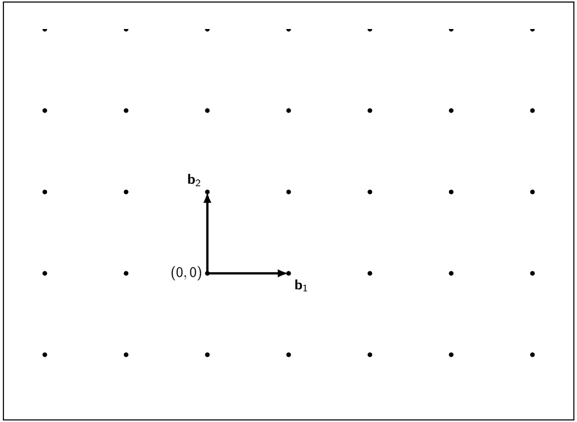
Defining a lattice

- Informally, lattices are a discrete equivalent of vector spaces



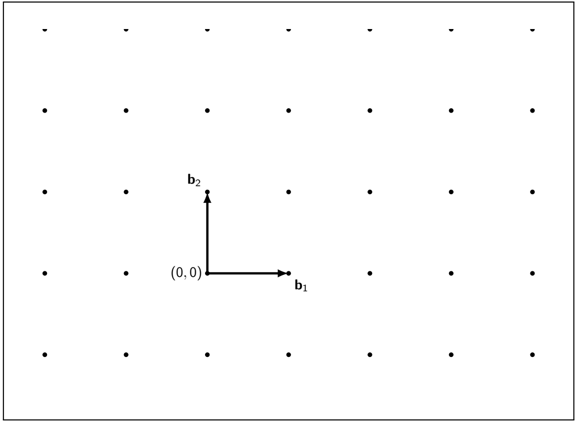
Defining a lattice

- Informally, lattices are a discrete equivalent of vector spaces
- The canonically simplest lattice is $\mathbb{Z}^n$.



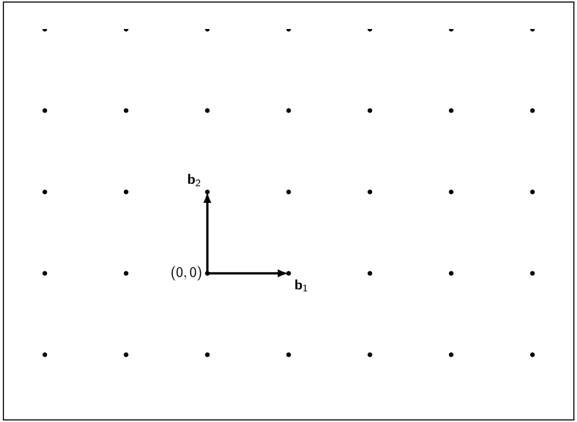
Defining a lattice

- Informally, lattices are a discrete equivalent of vector spaces
- The canonically simplest lattice is $\mathbb{Z}^n$.
- You can think of $\mathbb{Z}^n$ as the *integer span* of $(1, 0)$ and $(0, 1)$.



Defining a lattice

- Informally, lattices are a discrete equivalent of vector spaces
- The canonically simplest lattice is $\mathbb{Z}^n$.
- You can think of $\mathbb{Z}^n$ as the *integer span* of $(1, 0)$ and $(0, 1)$.
- Other lattices are *linear transformations* of $\mathbb{Z}^n$.



Defining a lattice

Definition

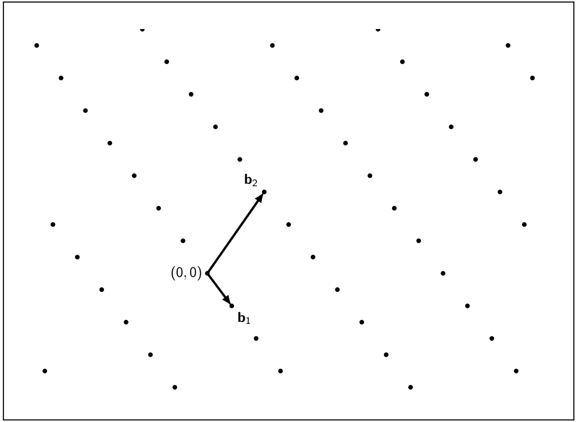
Let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^d$ be lin. indep.

$$\mathbf{B} := \begin{bmatrix} \text{---} \ \mathbf{b}_1 \ \text{---} \\ \vdots \\ \text{---} \ \mathbf{b}_n \ \text{---} \end{bmatrix}.$$

We say that their integer span

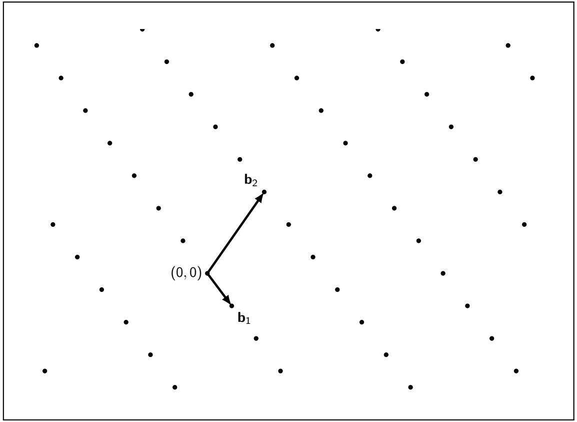
$$\begin{aligned} \Lambda = \Lambda(\mathbf{B}) &:= \mathrm{span}_{\mathbb{Z}}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \\ &= \{x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n : x_i \in \mathbb{Z}\} \\ &= \{\mathbf{xB} : \mathbf{x} \in \mathbb{Z}^n\} \subset \mathbb{R}^d, \end{aligned}$$

is a *real lattice* of rank, n.



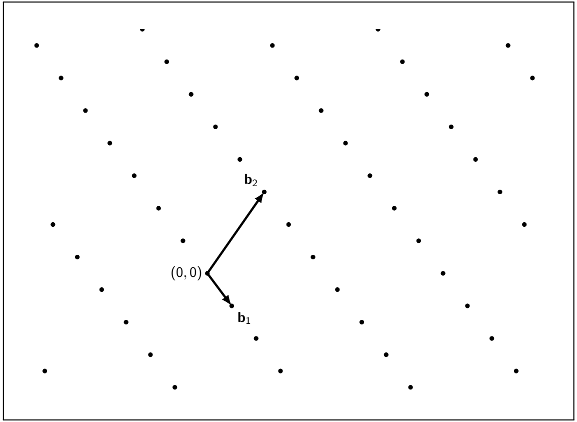
Defining a lattice

- If $n = d$ we say Λ is *full-rank*.



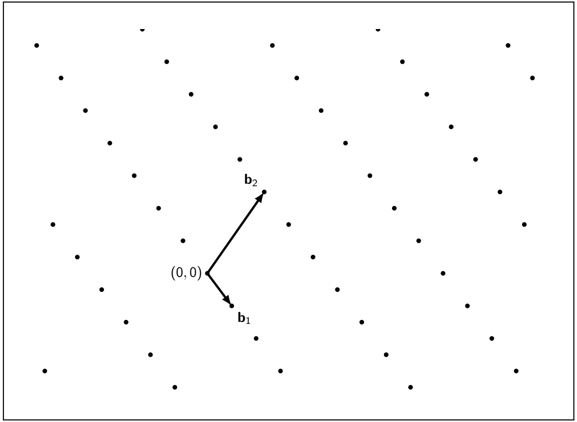
Defining a lattice

- If $n = d$ we say Λ is *full-rank*.
- If $\Lambda \subset \mathbb{Z}^d$, we say Λ is an *integer lattice*.



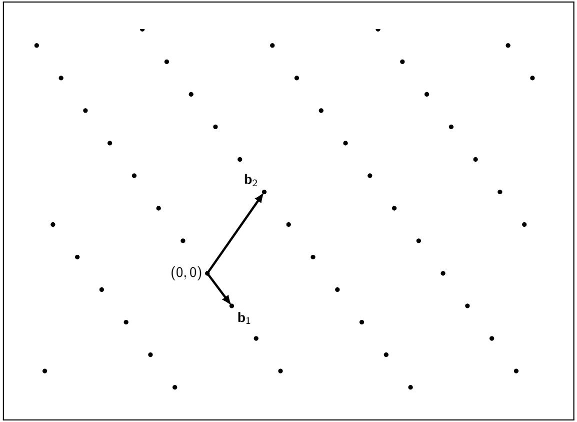
Defining a lattice

- If $n = d$ we say Λ is *full-rank*.
- If $\Lambda \subset \mathbb{Z}^d$, we say Λ is an *integer lattice*.
- Λ is a subgroup of $(\mathbb{R}^d, +)$



Defining a lattice

- If $n = d$ we say Λ is *full-rank*.
- If $\Lambda \subset \mathbb{Z}^d$, we say Λ is an *integer lattice*.
- Λ is a subgroup of $(\mathbb{R}^d, +)$
- Lattices are infinite sets, but we will want uniform distributions.

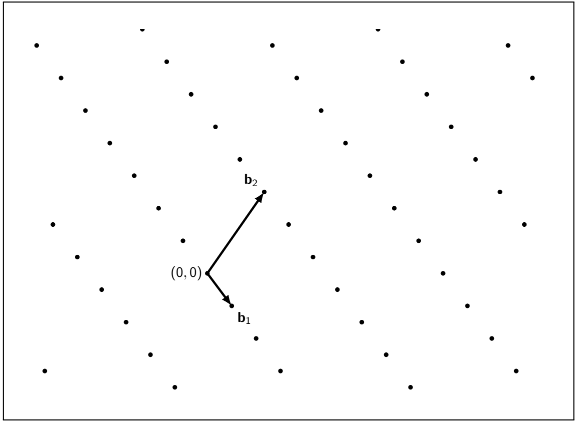


Defining a lattice

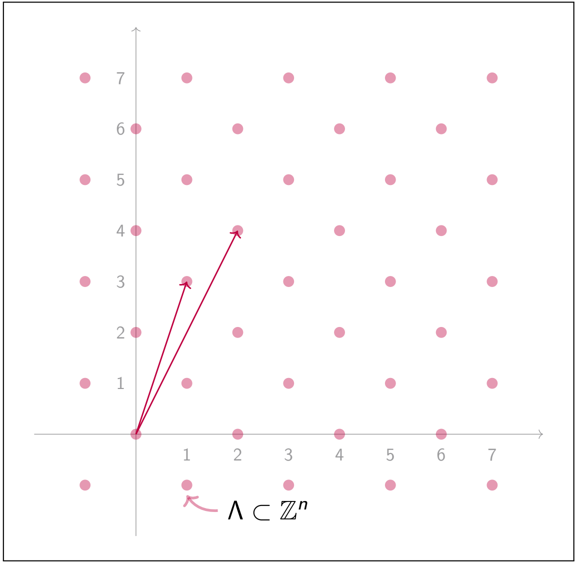
- If $n = d$ we say Λ is *full-rank*.
- If $\Lambda \subset \mathbb{Z}^d$, we say Λ is an *integer lattice*.
- Λ is a subgroup of $(\mathbb{R}^d, +)$
- Lattices are infinite sets, but we will want uniform distributions.

Definition (Sublattices)

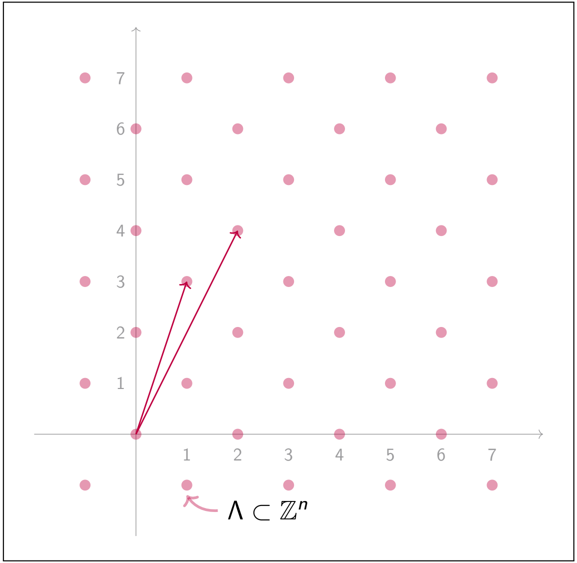
Let $\Lambda \subset \mathbb{R}^n$ be a real lattice of rank n. We call any subgroup $\Lambda' \subset \Lambda$ a sublattice of Λ.



- Lattices are infinite, making sampling uniformly difficult.



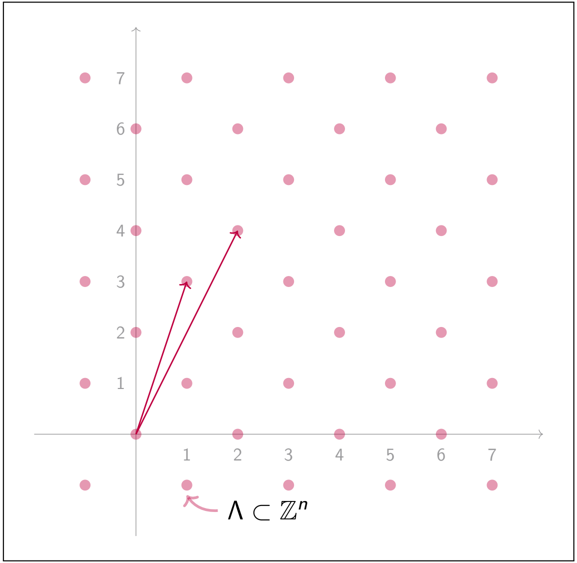
- Lattices are infinite, making sampling uniformly difficult.
- In cryptography, we address this by focusing on q-ary lattices.



- Lattices are infinite, making sampling uniformly difficult.
- In cryptography, we address this by focusing on q-ary lattices.

Definition

Λ is *q-ary* if $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$.

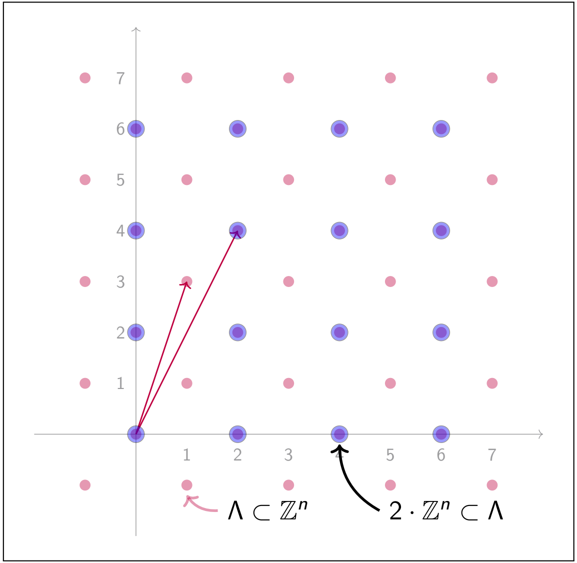


- Lattices are infinite, making sampling uniformly difficult.
- In cryptography, we address this by focusing on q-ary lattices.

Definition

Λ is *q-ary* if $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$.

- $q\mathbb{Z}^d$ is a subgroup of Λ.

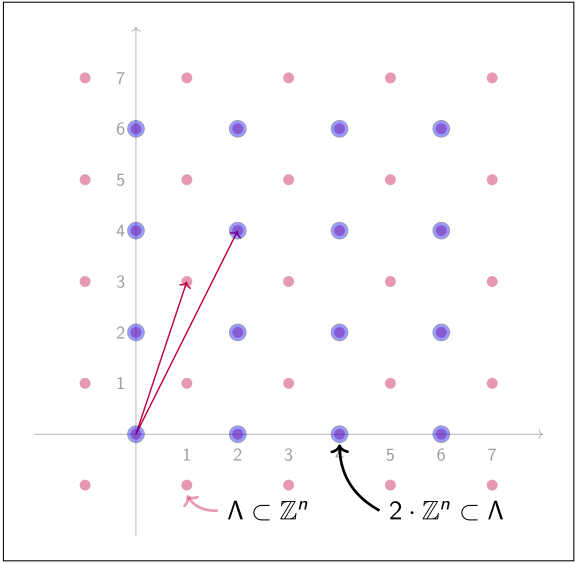


- Lattices are infinite, making sampling uniformly difficult.
- In cryptography, we address this by focusing on q-ary lattices.

Definition

Λ is *q-ary* if $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$.

- $q\mathbb{Z}^d$ is a subgroup of Λ.
- For any $\mathbf{v} \in \Lambda$, $\mathbf{v} + q\mathbb{Z}^d$ is a coset of Λ.

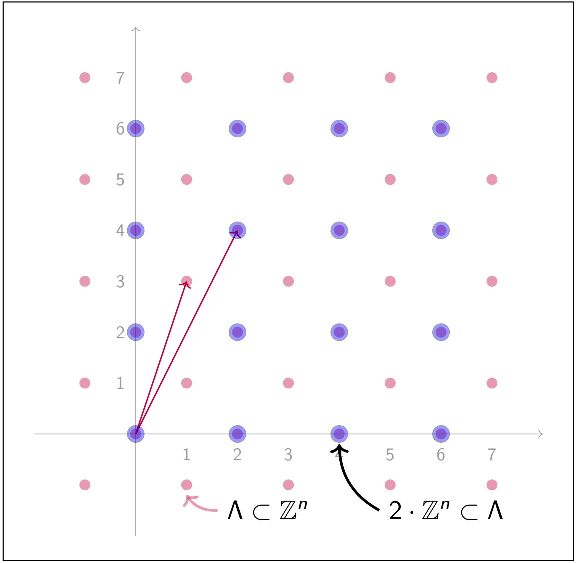


- Lattices are infinite, making sampling uniformly difficult.
- In cryptography, we address this by focusing on q-ary lattices.

Definition

Λ is *q-ary* if $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$.

- $q\mathbb{Z}^d$ is a subgroup of Λ.
- For any $\mathbf{v} \in \Lambda$, $\mathbf{v} + q\mathbb{Z}^d$ is a coset of Λ.
- Meaning $\Lambda \sim$ subgroup of $\mathbb{Z}_q^d$
  - $\mathbf{v} \mapsto \mathbf{v} \bmod q$

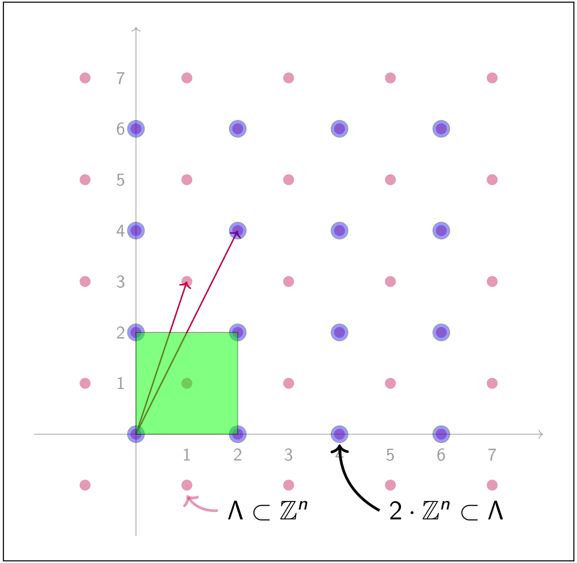


- Lattices are infinite, making sampling uniformly difficult.
- In cryptography, we address this by focusing on q-ary lattices.

Definition

Λ is *q-ary* if $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$.

- $q\mathbb{Z}^d$ is a subgroup of Λ.
- For any $\mathbf{v} \in \Lambda$, $\mathbf{v} + q\mathbb{Z}^d$ is a coset of Λ.
- Meaning $\Lambda \sim$ subgroup of $\mathbb{Z}_q^d$
  - $\mathbf{v} \mapsto \mathbf{v} \bmod q$
- This allows us to sample $U(\Lambda \bmod q)$.



Lattices have multiple bases.

Lattices have multiple bases.

Lemma

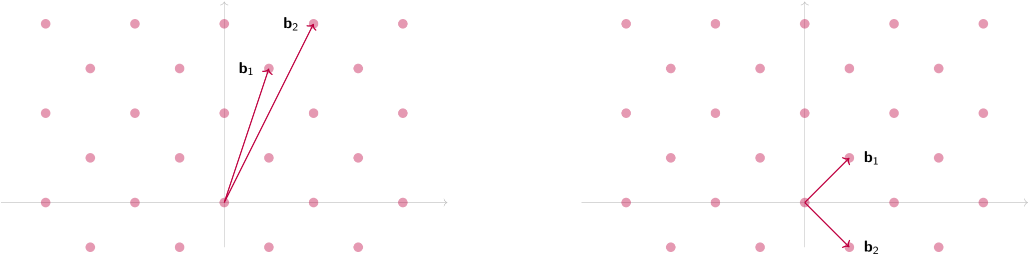
$\Lambda(\mathbf{B}) = \Lambda(\mathbf{B}')$ *if and only if* $\mathbf{B}' = \mathbf{U}\mathbf{B}$ *where* $\mathbf{U}$ *is unimodular* ($\mathbf{U} \in \mathbb{Z}^{n \times n}$ *with* $\det(\mathbf{U}) = \pm 1$) $\square$

Lattices have multiple bases.

Lemma

$\Lambda(\mathbf{B}) = \Lambda(\mathbf{B}')$ *if and only if* $\mathbf{B}' = \mathbf{U}\mathbf{B}$ *where* $\mathbf{U}$ *is unimodular* ($\mathbf{U} \in \mathbb{Z}^{n \times n}$ *with* $\det(\mathbf{U}) = \pm 1$) □

Unimodular matrices $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ give bijections $\mathbb{Z}^n \to \mathbb{Z}^n$ between coefficient vectors.



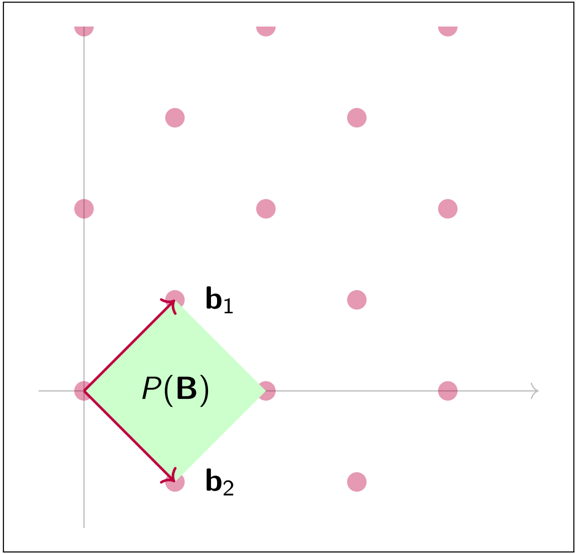
$$\mathbf{B} = \begin{bmatrix} (1, & 3) \\ (2, & 4) \end{bmatrix} \xmapsto{\mathbf{U} = \begin{bmatrix} -1 & 1 \\ -3 & 2 \end{bmatrix}} \mathbf{B}' = \begin{bmatrix} (1, & 1) \\ (1, & -1) \end{bmatrix}$$

Given a lattice basis, we can define its *fundamental parallelepiped* $P(\mathbf{B})$.

Definition

Given a lattice basis $\mathbf{B}$, its fundamental parallelepiped is the set

$$P(\mathbf{B}) := \{x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n \colon x_i \in [0,1)\}.$$

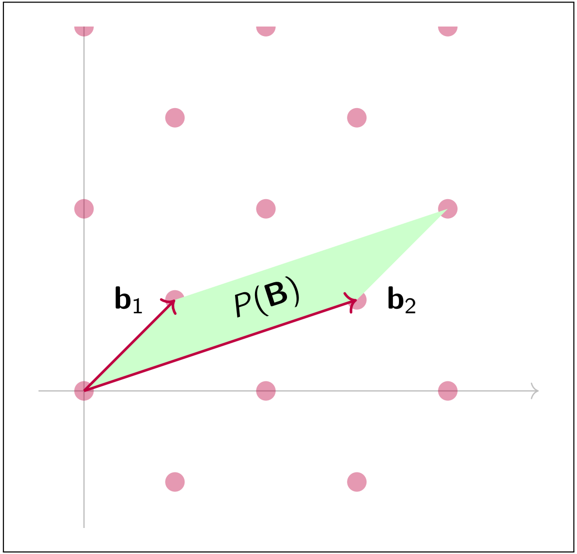


Given a lattice basis, we can define its *fundamental parallelepiped* $P(\mathbf{B})$.

Definition

Given a lattice basis $\mathbf{B}$, its fundamental parallelepiped is the set

$$P(\mathbf{B}) := \{x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n : x_i \in [0,1)\}.$$

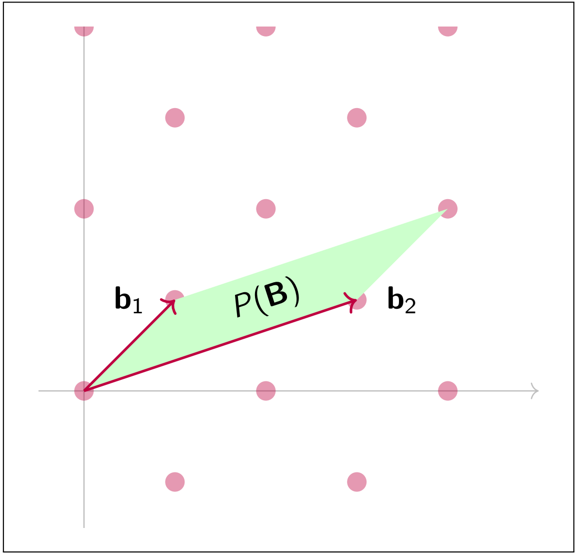


We now define a lattice invariant.

Definition

Given any basis $\mathbf{B}$ the *volume* of Λ is

$$\mathrm{vol}(\Lambda) := \sqrt{\det(\mathbf{B}\mathbf{B}^t)}$$



We now define a lattice invariant.

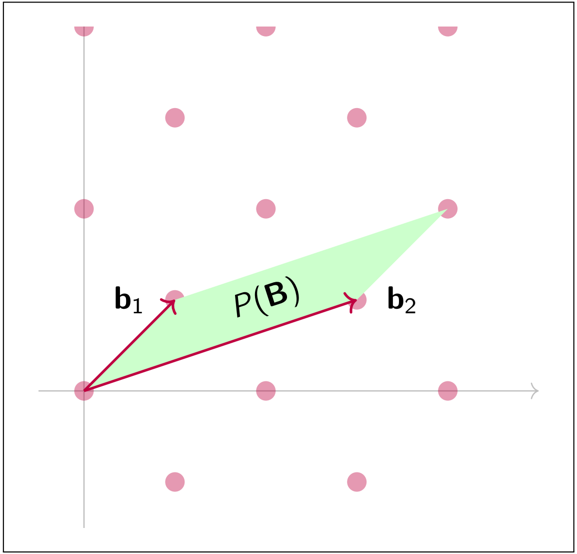
Definition

Given any basis $\mathbf{B}$ the *volume* of Λ is

$$\mathrm{vol}(\Lambda) := \sqrt{\det(\mathbf{B}\mathbf{B}^t)}$$

Lemma

- $\mathrm{Vol}(\Lambda) = \mathrm{Vol}(P(\Lambda)) = \int_{P(\Lambda)} d\mathbf{v}$
- *If* $\mathbf{B} \in \mathbb{Z}^{n\times n}$, *then* $\mathrm{Vol}(\Lambda) = |\det(\mathbf{B})|$



Sublattices may potentially have a smaller rank or a larger volume than Λ.

Sublattices may potentially have a smaller rank or a larger volume than Λ.

Example

Let $\mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{R}^n$ be the canonical basis of $\mathbb{R}^n$.
Let $\Lambda = \mathbb{Z}^n = \Lambda(\mathbf{e}_1, \ldots, \mathbf{e}_n)$.

Sublattices may potentially have a smaller rank or a larger volume than Λ.

Example

Let $\mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{R}^n$ be the canonical basis of $\mathbb{R}^n$.
Let $\Lambda = \mathbb{Z}^n = \Lambda(\mathbf{e}_1, \ldots, \mathbf{e}_n)$. Then

- $\Lambda(\mathbf{e}_2, \ldots, \mathbf{e}_n)$ is a sublattice of rank $n-1$ and volume 1.

Sublattices may potentially have a smaller rank or a larger volume than Λ.

Example

Let $\mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{R}^n$ be the canonical basis of $\mathbb{R}^n$.
Let $\Lambda = \mathbb{Z}^n = \Lambda(\mathbf{e}_1, \ldots, \mathbf{e}_n)$. Then

- $\Lambda(\mathbf{e}_2, \ldots, \mathbf{e}_n)$ is a sublattice of rank $n-1$ and volume 1.
- $\Lambda(2 \cdot \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$ is a sublattice of rank n and volume 2.

Sublattices may potentially have a smaller rank or a larger volume than Λ.

Example

Let $\mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{R}^n$ be the canonical basis of $\mathbb{R}^n$.
Let $\Lambda = \mathbb{Z}^n = \Lambda(\mathbf{e}_1, \ldots, \mathbf{e}_n)$. Then

- $\Lambda(\mathbf{e}_2, \ldots, \mathbf{e}_n)$ is a sublattice of rank $n-1$ and volume 1.
- $\Lambda(2 \cdot \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$ is a sublattice of rank n and volume 2.
- $\Lambda(2 \cdot \mathbf{e}_2, \mathbf{e}_3 \ldots, \mathbf{e}_n)$ is a sublattice of rank $n-1$ and volume 2.

Sublattices may potentially have a smaller rank or a larger volume than Λ.

Example

Let $\mathbf{e}_1, \ldots, \mathbf{e}_n \in \mathbb{R}^n$ be the canonical basis of $\mathbb{R}^n$.
Let $\Lambda = \mathbb{Z}^n = \Lambda(\mathbf{e}_1, \ldots, \mathbf{e}_n)$. Then

- $\Lambda(\mathbf{e}_2, \ldots, \mathbf{e}_n)$ is a sublattice of rank $n-1$ and volume 1.
- $\Lambda(2 \cdot \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$ is a sublattice of rank n and volume 2.
- $\Lambda(2 \cdot \mathbf{e}_2, \mathbf{e}_3 \ldots, \mathbf{e}_n)$ is a sublattice of rank $n-1$ and volume 2.

WLOG, we can choose to work only with unit-volume lattices.

- Any lattice keeps the same "structure" when scaled down by $\mathrm{Vol}(\Lambda)^{1/n}$ in all directions.

Successive minima

Successive minima

Definition

Let $B_d(r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| \leq r\}$ be the closed ball of radius r in $\mathbb{R}^d$ centered around 0.

Successive minima

Definition

Let $B_d(r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| \leq r\}$ be the closed ball of radius r in $\mathbb{R}^d$ centered around 0. We define the *i-th minima of* Λ as

$$\lambda_i(\Lambda) = \min\left\{r \in \mathbb{R}^+ \colon \Lambda \cap B_d(r) \text{ contains } i \text{ linearly independent vectors}\right\}.$$

Successive minima

Definition

Let $B_d(r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| \leq r\}$ be the closed ball of radius r in $\mathbb{R}^d$ centered around 0.
We define the *i-th minima of* Λ as

$$\lambda_i(\Lambda) = \min\left\{r \in \mathbb{R}^+ : \Lambda \cap B_d(r) \text{ contains } i \text{ linearly independent vectors}\right\}.$$

Minima are norms, not vectors!

Successive minima

Definition

Let $B_d(r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| \leq r\}$ be the closed ball of radius r in $\mathbb{R}^d$ centered around 0.
We define the *i-th minima of* Λ as

$$\lambda_i(\Lambda) = \min\left\{r \in \mathbb{R}^+ : \Lambda \cap B_d(r) \text{ contains } i \text{ linearly independent vectors}\right\}.$$

Minima are norms, not vectors!

$$\lambda_1(\Lambda) \leq \lambda_2(\Lambda) \leq \cdots \leq \lambda_n(\Lambda)$$

Successive minima

Definition

Let $B_d(r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| \leq r\}$ be the closed ball of radius r in $\mathbb{R}^d$ centered around 0.
We define the *i-th minima of* Λ as

$$\lambda_i(\Lambda) = \min \left\{ r \in \mathbb{R}^+ : \Lambda \cap B_d(r) \text{ contains } i \text{ linearly independent vectors} \right\}.$$

Minima are norms, not vectors!

$$\lambda_1(\Lambda) \leq \lambda_2(\Lambda) \leq \cdots \leq \lambda_n(\Lambda)$$

Example

- $\Lambda = \mathbb{Z} \times \mathbb{Z}$ has $\lambda_1(\Lambda) = \lambda_2(\Lambda) = 1$.

Successive minima

Definition

Let $B_d(r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| \leq r\}$ be the closed ball of radius r in $\mathbb{R}^d$ centered around 0.
We define the *i-th minima of* Λ as

$$\lambda_i(\Lambda) = \min\left\{r \in \mathbb{R}^+ : \Lambda \cap B_d(r) \text{ contains } i \text{ linearly independent vectors}\right\}.$$

Minima are norms, not vectors!

$$\lambda_1(\Lambda) \leq \lambda_2(\Lambda) \leq \cdots \leq \lambda_n(\Lambda)$$

Example

- $\Lambda = \mathbb{Z} \times \mathbb{Z}$ has $\lambda_1(\Lambda) = \lambda_2(\Lambda) = 1$.
- $\Lambda = \mathbb{Z} \times (2\mathbb{Z})$ has $\lambda_1(\Lambda) = 1$ and $\lambda_2(\Lambda) = 2$.

Estimating λ_1: worst-case.

Estimating λ_1: worst-case.

Definition (Hermite's constant [3])

Let $\mathcal{L}_n$ be the set of real lattices of rank n. Then Hermite's constant for rank n lattices, γ_n, is

$$\gamma_n := \sup_{\Lambda \in \mathcal{L}_n} \frac{\lambda_1(\Lambda)^2}{\mathrm{vol}(\Lambda)^{2/n}}.$$

Estimating λ_1: worst-case.

Definition (Hermite's constant [3])

Let $\mathcal{L}_n$ be the set of real lattices of rank n. Then Hermite's constant for rank n lattices, γ_n, is

$$\gamma_n := \sup_{\Lambda \in \mathcal{L}_n} \frac{\lambda_1(\Lambda)^2}{\mathrm{vol}(\Lambda)^{2/n}}.$$

Hermite's constant is known for $n \leq 8$ and $n = 24$ [4, § 6].

- Eg, $\gamma_2 = \sqrt{4/3}$

Estimating λ_1: worst-case.

Definition (Hermite's constant [3])

Let $\mathcal{L}_n$ be the set of real lattices of rank n. Then Hermite's constant for rank n lattices, γ_n, is

$$\gamma_n := \sup_{\Lambda \in \mathcal{L}_n} \frac{\lambda_1(\Lambda)^2}{\mathrm{vol}(\Lambda)^{2/n}}.$$

Hermite's constant is known for $n \leq 8$ and $n = 24$ [4, § 6].

- Eg, $\gamma_2 = \sqrt{4/3}$

We know upper bounds.

Estimating λ_1: worst-case.

Definition (Hermite's constant [3])

Let $\mathcal{L}_n$ be the set of real lattices of rank n. Then Hermite's constant for rank n lattices, γ_n, is

$$\gamma_n := \sup_{\Lambda \in \mathcal{L}_n} \frac{\lambda_1(\Lambda)^2}{\mathrm{vol}(\Lambda)^{2/n}}.$$

Hermite's constant is known for $n \leq 8$ and $n = 24$ [4, § 6].

- Eg, $\gamma_2 = \sqrt{4/3}$

We know upper bounds.

Theorem (Hermite's inequality [3])

Let $n \geq 2$ be an integer. Then $\gamma_n \leq \gamma_2^{n-1}$.

Estimating λ_1: worst-case.

Definition (Hermite's constant [3])

Let $\mathcal{L}_n$ be the set of real lattices of rank n. Then Hermite's constant for rank n lattices, γ_n, is

$$\gamma_n := \sup_{\Lambda \in \mathcal{L}_n} \frac{\lambda_1(\Lambda)^2}{\mathrm{vol}(\Lambda)^{2/n}}.$$

Hermite's constant is known for $n \leq 8$ and $n = 24$ [4, § 6].

- Eg, $\gamma_2 = \sqrt{4/3}$

We know upper bounds.

Theorem (Hermite's inequality [3])

Let $n \geq 2$ be an integer. Then $\gamma_n \leq \gamma_2^{n-1}$.

Corollary

Given any lattice Λ of rank n, it contains $\mathbf{v} \neq \mathbf{0}$ of norm

$$\|\mathbf{v}\| \leq \sqrt{\gamma_n} \cdot \mathrm{vol}(\Lambda)^{1/n} \leq \gamma_2^{(n-1)/2} \cdot \mathrm{vol}(\Lambda)^{1/n}.$$

Estimating λ_1: worst-case.

Definition (Hermite's constant [3])

Let $\mathcal{L}_n$ be the set of real lattices of rank n. Then Hermite's constant for rank n lattices, γ_n, is

$$\gamma_n := \sup_{\Lambda \in \mathcal{L}_n} \frac{\lambda_1(\Lambda)^2}{\mathrm{vol}(\Lambda)^{2/n}}.$$

Hermite's constant is known for $n \leq 8$ and $n = 24$ [4, § 6].

- Eg, $\gamma_2 = \sqrt{4/3}$

We know upper bounds.

Theorem (Hermite's inequality [3])

Let $n \geq 2$ be an integer. Then $\gamma_n \leq \gamma_2^{n-1}$.

Corollary

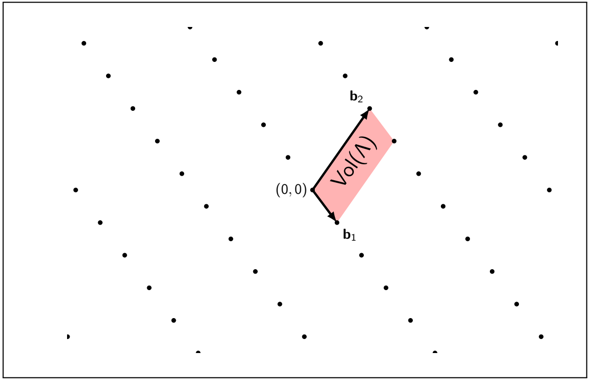
Given any lattice Λ of rank n, it contains $\mathbf{v} \neq \mathbf{0}$ of norm

$$\|\mathbf{v}\| \leq \sqrt{\gamma_n} \cdot \mathrm{vol}(\Lambda)^{1/n} \leq \gamma_2^{(n-1)/2} \cdot \mathrm{vol}(\Lambda)^{1/n}.$$

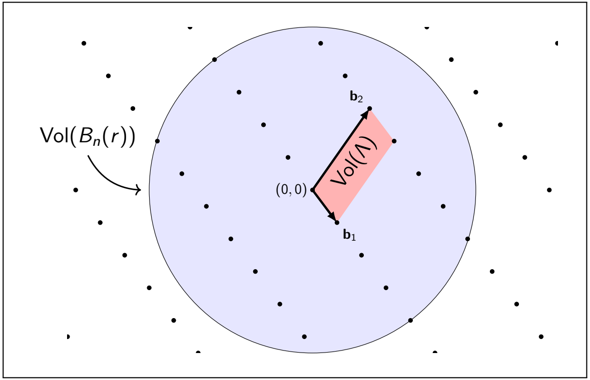
Theorem (Mordell's inequality [5])

Let $n \geq k \geq 2$ be integers. Then $\gamma_n^{k-1} \leq \gamma_k^{n-1}$.

Estimating λ_1: average-case.



Estimating λ_1: average-case.

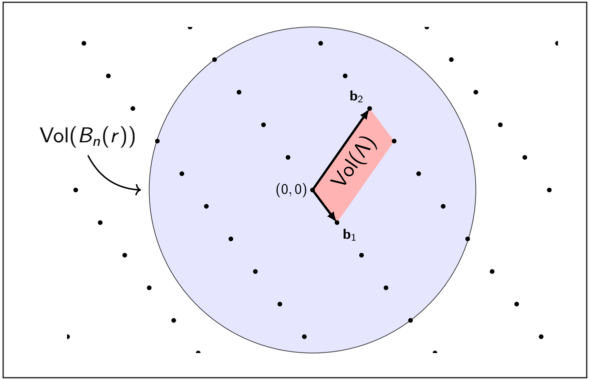


Estimating λ_1: average-case.

Heuristic (Gaussian heuristic)

Let $S \in \mathrm{span}(\mathbf{B})$ *be a measurable set.*

Then $$\#\Lambda \cap S \approx \frac{\mathrm{Vol}(S)}{\mathrm{Vol}(\Lambda)}$$



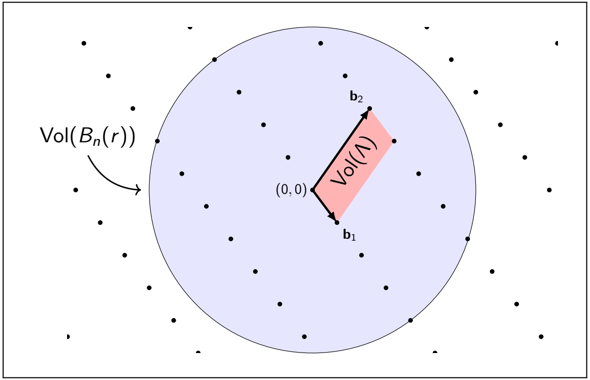
Estimating λ_1: average-case.

Heuristic (Gaussian heuristic)

Let $S \in \text{span}(\mathbf{B})$ *be a measurable set.*
Then $$\#\Lambda \cap S \approx \frac{\text{Vol}(S)}{\text{Vol}(\Lambda)}$$

Heuristic (Gaussian heuristic for λ_1)

Let $S = B_n(r)$ *such that* $\text{Vol}(S) = \text{Vol}(\Lambda)$. *Then*



Estimating λ_1: average-case.

Heuristic (Gaussian heuristic)

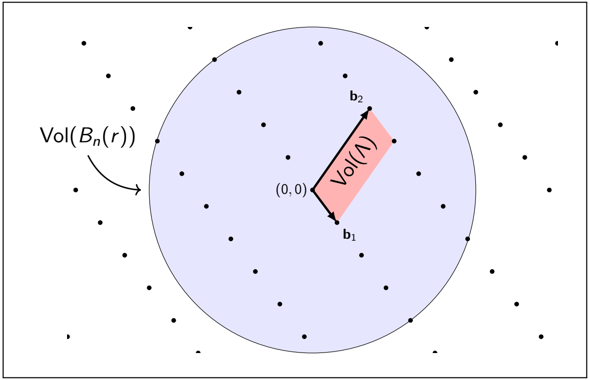
Let $S \in \mathrm{span}(\mathbf{B})$ *be a measurable set.*

Then $\#\Lambda \cap S \approx \frac{\mathrm{Vol}(S)}{\mathrm{Vol}(\Lambda)}$

Heuristic (Gaussian heuristic for λ_1)

Let $S = B_n(r)$ *such that* $\mathrm{Vol}(S) = \mathrm{Vol}(\Lambda)$. *Then*

$$\lambda_1(\Lambda) \approx r$$



Estimating λ_1: average-case.

Heuristic (Gaussian heuristic)

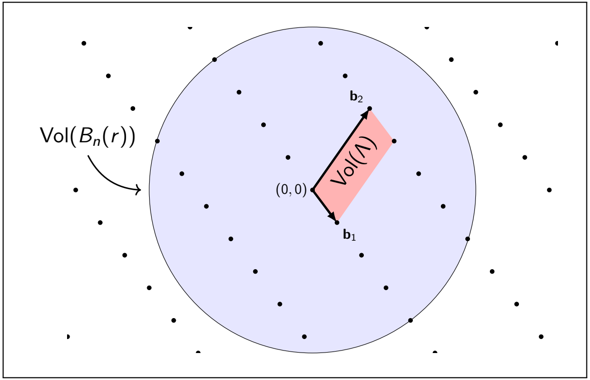
Let $S \in \mathrm{span}(\mathbf{B})$ *be a measurable set.*

Then $\#\Lambda \cap S \approx \frac{\mathrm{Vol}(S)}{\mathrm{Vol}(\Lambda)}$

Heuristic (Gaussian heuristic for λ_1)

Let $S = B_n(r)$ *such that* $\mathrm{Vol}(S) = \mathrm{Vol}(\Lambda)$. *Then*

$$\lambda_1(\Lambda) \approx r = \frac{\Gamma(1+n/2)^{1/n}}{\sqrt{\pi}} \mathrm{vol}(\Lambda)^{1/n}$$



Estimating λ_1: average-case.

Heuristic (Gaussian heuristic)

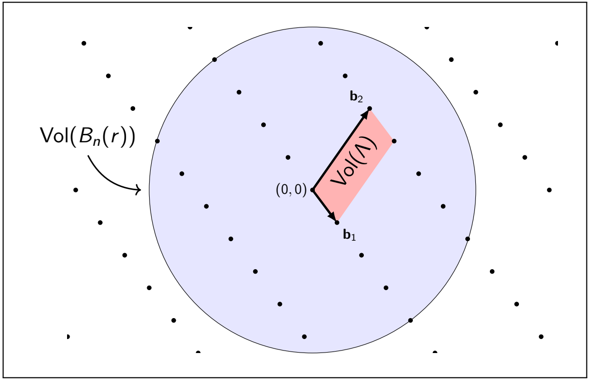
Let $S \in \text{span}(\mathbf{B})$ *be a measurable set.*
Then $\#\Lambda \cap S \approx \frac{\text{Vol}(S)}{\text{Vol}(\Lambda)}$

Heuristic (Gaussian heuristic for λ_1)

Let $S = B_n(r)$ *such that* $\text{Vol}(S) = \text{Vol}(\Lambda)$. *Then*

$$\lambda_1(\Lambda) \approx r = \frac{\Gamma(1+n/2)^{1/n}}{\sqrt{\pi}} \mathrm{vol}(\Lambda)^{1/n}$$

$$\approx (\pi n)^{\frac{1}{2n}} \sqrt{\frac{n}{2\pi e}} \mathrm{vol}(\Lambda)^{1/n} \quad \textit{by [6, §II.9].}$$



Estimating λ_1: average-case.

Heuristic (Gaussian heuristic)

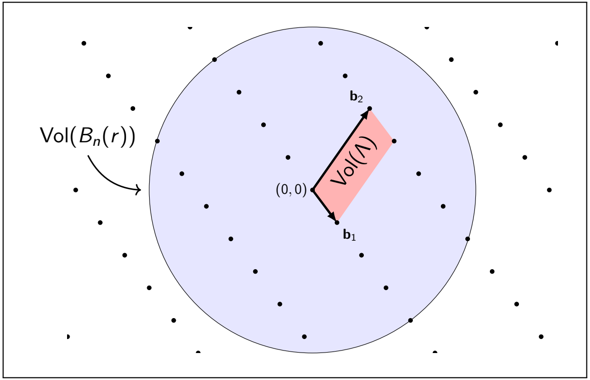
Let $S \in \mathrm{span}(\mathbf{B})$ *be a measurable set.*
Then $\#\Lambda \cap S \approx \frac{\mathrm{Vol}(S)}{\mathrm{Vol}(\Lambda)}$

Heuristic (Gaussian heuristic for λ_1)

Let $S = B_n(r)$ *such that* $\mathrm{Vol}(S) = \mathrm{Vol}(\Lambda)$. *Then*

$$\lambda_1(\Lambda) \approx r = \frac{\Gamma(1+n/2)^{1/n}}{\sqrt{\pi}} \mathrm{vol}(\Lambda)^{1/n}$$

$$\approx (\pi n)^{\frac{1}{2n}} \sqrt{\frac{n}{2\pi e}} \mathrm{vol}(\Lambda)^{1/n} \quad \textit{by [6, §II.9].}$$



While estimating λ_1 is generally easy, finding a vector realising λ_1 is generally hard!

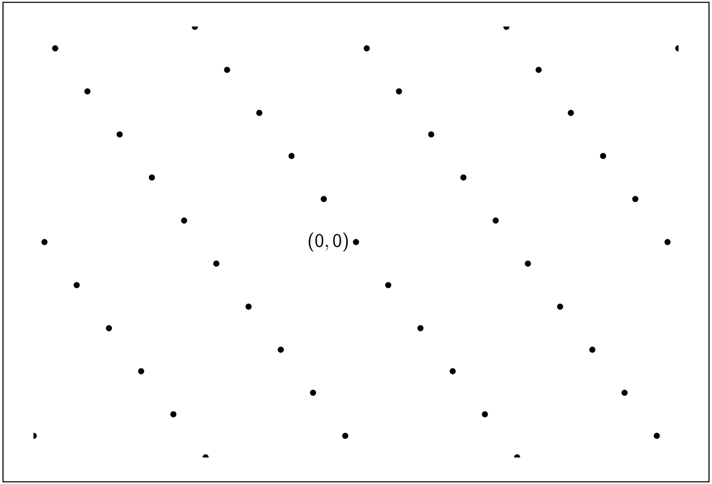
Computational problems and hardness assumptions

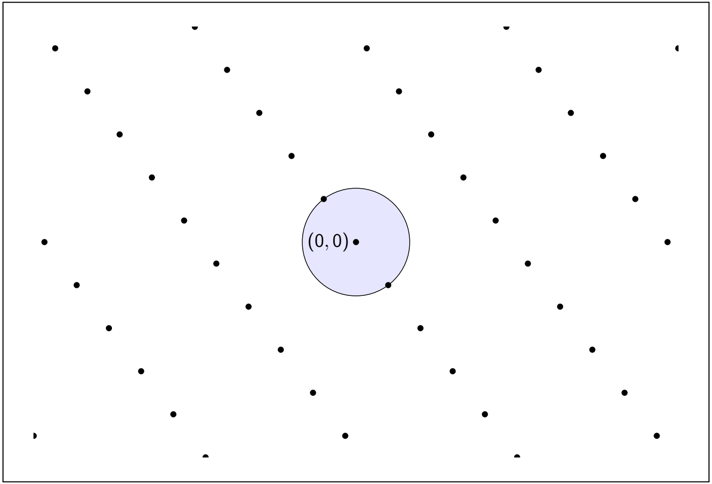
Fernando Virdia — `https://fundamental.domains`
EPFL-ETH Summer School on Lattice-based Cryptography, July 2025

- We've looked at lattices as mathematical objects.

- We've looked at lattices as mathematical objects.
- What about lattices as sources of computational problems?

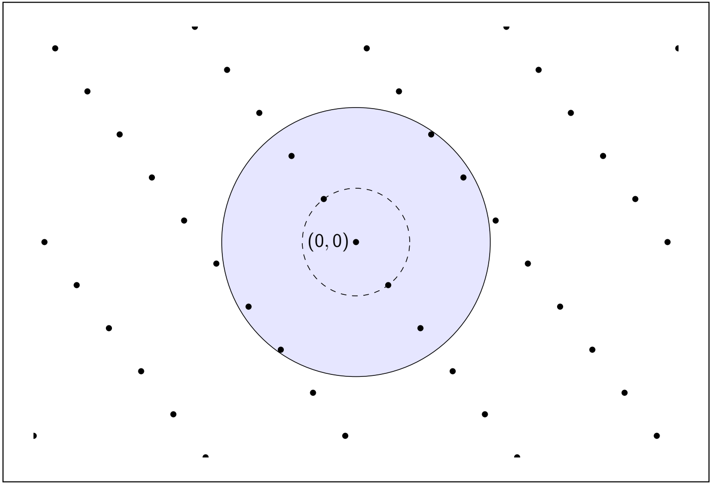
- We've looked at lattices as mathematical objects.
- What about lattices as sources of computational problems?
- Various questions can be asked of a lattice, many giving rise to problems hard in the worst case.


(0, 0)



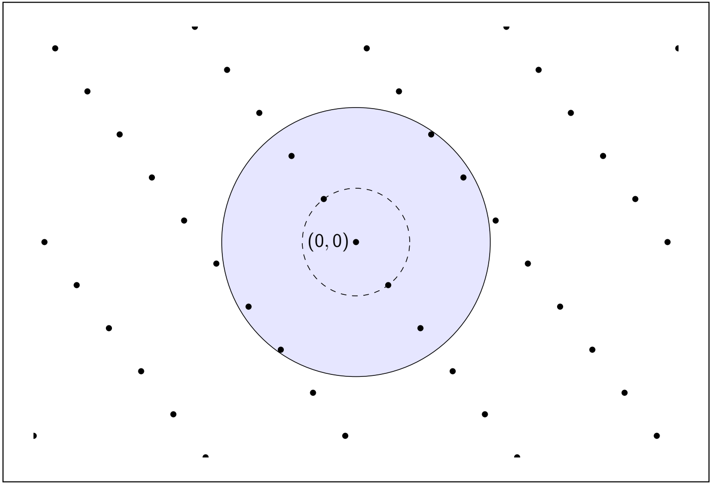
Definition (Shortest Vector Problem (SVP))

Given a lattice Λ find a vector $\mathbf{v} \in \Lambda$ of norm $\lambda_1(\Lambda)$.



Definition (γ-approximate Shortest Vector Problem (approx-SVP$_\gamma$))

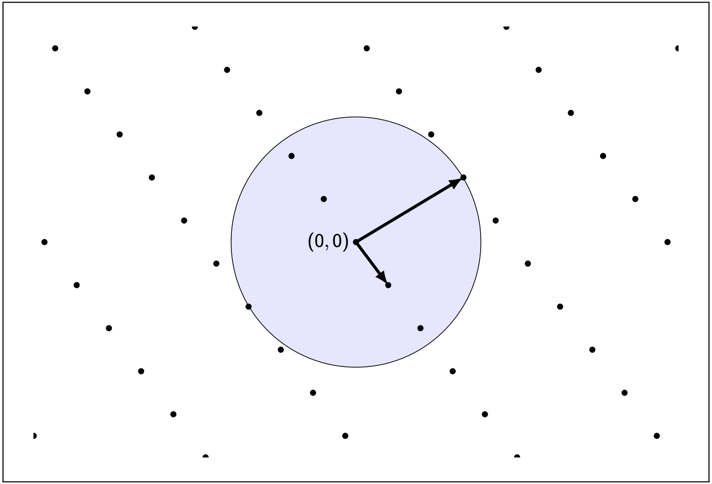
Given a lattice Λ, find a non-zero vector $\mathbf{v} \in \Lambda$ of norm $\leq \gamma \cdot \lambda_1(\Lambda)$.



Definition (γ-Hermite Shortest Vector Problem (Hermite-SVP$_\gamma$))

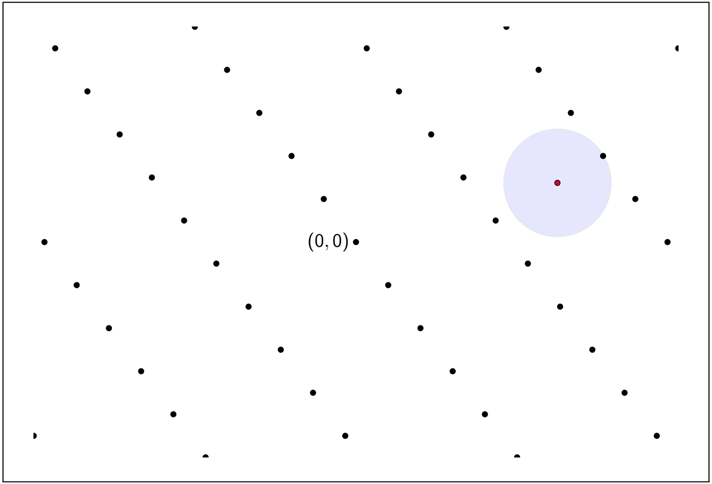
Given a lattice Λ, find a non-zero vector $\mathbf{v} \in \Lambda$ of norm $\leq \gamma \cdot \mathrm{vol}(\Lambda)^{1/n}$.

Approx-SVP is *relative* to Λ, Hermite-SVP is *absolute*: every Λ can be scaled to $\mathrm{Vol}(\Lambda) = 1$.



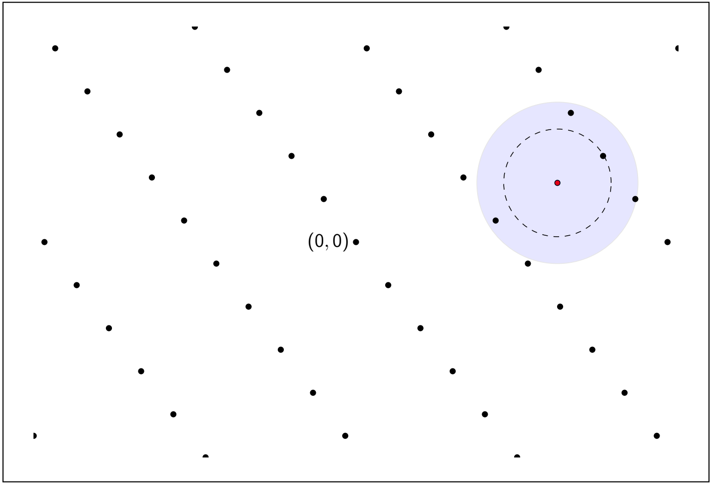
Definition (γ-Shortest Independent Vectors Problem (SIVP_γ))

Given a lattice Λ of rank n, find n linearly independent lattice vectors $\mathbf{v}_i \in \Lambda$ of norm at most $\gamma \cdot \lambda_n(\Lambda)$.



Definition (γ-Closest Vector Problem (CVP$_\gamma$))

Given a lattice basis $\mathbf{B}$ and a vector $\mathbf{v} \in \mathrm{span}_{\mathbb{R}}(\mathbf{B})$, find a lattice point $\mathbf{u} \in \Lambda(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{u}\| \leq \gamma \cdot \min_{\mathbf{x} \in \Lambda(\mathbf{B})} \|\mathbf{v} - \mathbf{x}\|$.



Definition (γ-Closest Vector Problem (CVP_γ))

Given a lattice basis $\mathbf{B}$ and a vector $\mathbf{v} \in \mathrm{span}_{\mathbb{R}}(\mathbf{B})$, find a lattice point $\mathbf{u} \in \Lambda(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{u}\| \leq \gamma \cdot \min_{\mathbf{x} \in \Lambda(\mathbf{B})} \|\mathbf{v} - \mathbf{x}\|$.

- The SVP and CVP problems also have “promise” variants.

- The SVP and CVP problems also have “promise” variants.

Definition (γ-unique Shortest Vector Problem (uSVP$_\gamma$))

Given a lattice Λ such that $\lambda_2(\Lambda) > \gamma \cdot \lambda_1(\Lambda)$, find the unique (up to sign) vector $\mathbf{v} \in \Lambda$ of norm $\lambda_1(\Lambda)$. Unless specified, $\gamma = 1$.

- The SVP and CVP problems also have "promise" variants.

Definition (γ-unique Shortest Vector Problem (uSVP_γ))

Given a lattice Λ such that $\lambda_2(\Lambda) > \gamma \cdot \lambda_1(\Lambda)$, find the unique (up to sign) vector $\mathbf{v} \in \Lambda$ of norm $\lambda_1(\Lambda)$. Unless specified, $\gamma = 1$.

Definition (γ-Bounded Distance Decoding (BDD_γ))

Given a lattice Λ and a vector $\mathbf{v} \in \mathrm{span}_{\mathbb{R}}(\Lambda) \setminus \Lambda$ such that

$$\mathrm{dist}(\mathbf{v}, \Lambda) := \min_{\mathbf{x} \in \Lambda} \|\mathbf{v} - \mathbf{x}\| < \gamma \cdot \lambda_1(\Lambda),$$

find $\mathbf{t} \in \Lambda$ such that $\|\mathbf{v} - \mathbf{t}\| = \mathrm{dist}(\mathbf{v}, \Lambda)$.

Note: $\mathbf{t}$ is unique (up to $\pm$) if $\gamma < 1/2$.

- The SVP and CVP problems also have "promise" variants.

Definition (γ-unique Shortest Vector Problem (uSVP$_\gamma$))

Given a lattice Λ such that $\lambda_2(\Lambda) > \gamma \cdot \lambda_1(\Lambda)$, find the unique (up to sign) vector $\mathbf{v} \in \Lambda$ of norm $\lambda_1(\Lambda)$. Unless specified, $\gamma = 1$.

Definition (γ-Bounded Distance Decoding (BDD$_\gamma$))

Given a lattice Λ and a vector $\mathbf{v} \in \mathrm{span}_{\mathbb{R}}(\Lambda) \setminus \Lambda$ such that

$$\mathrm{dist}(\mathbf{v}, \Lambda) := \min_{\mathbf{x} \in \Lambda} \|\mathbf{v} - \mathbf{x}\| < \gamma \cdot \lambda_1(\Lambda),$$

find $\mathbf{t} \in \Lambda$ such that $\|\mathbf{v} - \mathbf{t}\| = \mathrm{dist}(\mathbf{v}, \Lambda)$.

Note: $\mathbf{t}$ is unique (up to $\pm$) if $\gamma < 1/2$.

- These are relevant when a short vector is "planted" into a lattice.

Computational problems vs hardness assumptions

- For the classes of lattices used in cryptography, these problems are believed to be hard *in the worst case*,

Computational problems vs hardness assumptions

- For the classes of lattices used in cryptography, these problems are believed to be hard *in the worst case*,
- But hardness assumptions need to be hard *on average*!

Computational problems vs hardness assumptions

- For the classes of lattices used in cryptography, these problems are believed to be hard *in the worst case*,
- But hardness assumptions need to be hard *on average*!
- When constructing primitives, we (I?) don't think in terms of SIVP.

Computational problems vs hardness assumptions

- For the classes of lattices used in cryptography, these problems are believed to be hard *in the worst case*,
- But hardness assumptions need to be hard *on average*!
- When constructing primitives, we (I?) don't think in terms of SIVP.
- Rather, we usually rely on two (families) of assumptions:

Computational problems vs hardness assumptions

- For the classes of lattices used in cryptography, these problems are believed to be hard *in the worst case*,
- But hardness assumptions need to be hard *on average*!
- When constructing primitives, we (I?) don't think in terms of SIVP.
- Rather, we usually rely on two (families) of assumptions:

In the *minicrypt* corner

The Short Integer Solution (SIS) problem

Computational problems vs hardness assumptions

- For the classes of lattices used in cryptography, these problems are believed to be hard *in the worst case*,
- But hardness assumptions need to be hard *on average*!
- When constructing primitives, we (I?) don't think in terms of SIVP.
- Rather, we usually rely on two (families) of assumptions:

In the *minicrypt* corner	In the *cryptomania* corner
The Short Integer Solution (SIS) problem	The Learning With Errors (LWE) problem

Short Integer Solution problem

- Let $n, m, q \in \mathbb{N}$ and $B > 0$.

Short Integer Solution problem

- Let $n, m, q \in \mathbb{N}$ and $B > 0$.
- Let $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$.

$$\boxed{\quad \mathbf{A} \quad}$$

Short Integer Solution problem

- Let $n, m, q \in \mathbb{N}$ and $B > 0$.
- Let $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$.
- Let $\mathbf{b} \in \mathbb{Z}_q^n$.

$$\boxed{\quad \mathbf{A} \quad} = \boxed{\mathbf{b}} \mod q,$$

Short Integer Solution problem

- Let $n, m, q \in \mathbb{N}$ and $B > 0$.
- Let $\mathbf{A} \sim U(\mathbb{Z}_q^{n\times m})$.
- Let $\mathbf{b} \in \mathbb{Z}_q^n$.

$$\mathbf{A} \, \mathbf{x} = \mathbf{b} \mod q,$$

Definition

$\mathsf{SIS}_{n,m,q,B,p}$ is the problem of recovering an integer solution $\mathbf{x} \in \mathbb{Z}^m$ with $\|\mathbf{x}\|_p \leq B$.

Short Integer Solution problem

- Let $n, m, q \in \mathbb{N}$ and $B > 0$.
- Let $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$.
- Let $\mathbf{b} \in \mathbb{Z}_q^n$.

$$\boxed{\quad \mathbf{A} \quad} \; \boxed{\mathbf{x}} = \boxed{\mathbf{b}} \mod q,$$

Definition

$\mathsf{SIS}_{n,m,q,B,p}$ is the problem of recovering an integer solution $\mathbf{x} \in \mathbb{Z}^m$ with $\|\mathbf{x}\|_p \le B$.

Homogeneous SIS is the SIS problem when $\mathbf{b} = \mathbf{0}$.

Inhomogeneous SIS (I-SIS) is the SIS problem when $\mathbf{b} \xleftarrow{\$} U(\mathbb{Z}_q^n)$.

Most commonly, $p = 2$ or $p = \infty$.

Cryptographic application: collision-resistant hashing.

Cryptographic application: collision-resistant hashing.

Consider a family of functions $f_{\mathbf{A}} \colon \{0, 1, \ldots, B\}^m \to \mathbb{Z}_q^n$ given by

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q, \quad \text{where} \quad \mathbf{A} \in \mathbb{Z}_q^{n \times m}.$$

Cryptographic application: collision-resistant hashing.

Consider a family of functions $f_{\mathbf{A}} \colon \{0, 1, \ldots, B\}^m \to \mathbb{Z}_q^n$ given by

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q, \quad \text{where} \quad \mathbf{A} \in \mathbb{Z}_q^{n \times m}.$$

- Let $m \cdot \log(B + 1) > n \log q$, so that $f_{\mathbf{A}}$ is *compressing*.

Cryptographic application: collision-resistant hashing.

Consider a family of functions $f_{\mathbf{A}}\colon \{0, 1, \ldots, B\}^m \to \mathbb{Z}_q^n$ given by

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q, \quad \text{where} \quad \mathbf{A} \in \mathbb{Z}_q^{n \times m}.$$

- Let $m \cdot \log(B+1) > n \log q$, so that $f_{\mathbf{A}}$ is *compressing*.
- By sampling $\mathbf{A}$ uniformly at random, $f_{\mathbf{A}}$ becomes a *keyed hash function*.

Cryptographic application: collision-resistant hashing.

Consider a family of functions $f_{\mathbf{A}}\colon \{0, 1, \ldots, B\}^m \to \mathbb{Z}_q^n$ given by

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q, \quad \text{where} \quad \mathbf{A} \in \mathbb{Z}_q^{n\times m}.$$

- Let $m \cdot \log(B+1) > n \log q$, so that $f_{\mathbf{A}}$ is *compressing*.
- By sampling $\mathbf{A}$ uniformly at random, $f_{\mathbf{A}}$ becomes a *keyed hash function*.

Lemma

$f_{\mathbf{A}}$ is collision-resistance under (homogeneous) SIS with $p = \infty$.

Cryptographic application: collision-resistant hashing.

Consider a family of functions $f_{\mathbf{A}} \colon \{0, 1, \ldots, B\}^m \to \mathbb{Z}_q^n$ given by

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q, \quad \text{where} \quad \mathbf{A} \in \mathbb{Z}_q^{n \times m}.$$

- Let $m \cdot \log(B + 1) > n \log q$, so that $f_{\mathbf{A}}$ is *compressing*.
- By sampling $\mathbf{A}$ uniformly at random, $f_{\mathbf{A}}$ becomes a *keyed hash function*.

Lemma

$f_{\mathbf{A}}$ is collision-resistance under (homogeneous) SIS with $p = \infty$.

Proof.

Suppose $\mathbf{x} \neq \mathbf{x}'$ are a collision.

Cryptographic application: collision-resistant hashing.

Consider a family of functions $f_{\mathbf{A}} \colon \{0, 1, \ldots, B\}^m \to \mathbb{Z}_q^n$ given by

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q, \quad \text{where} \quad \mathbf{A} \in \mathbb{Z}_q^{n \times m}.$$

- Let $m \cdot \log(B+1) > n \log q$, so that $f_{\mathbf{A}}$ is *compressing*.
- By sampling $\mathbf{A}$ uniformly at random, $f_{\mathbf{A}}$ becomes a *keyed hash function*.

Lemma

$f_{\mathbf{A}}$ is collision-resistance under (homogeneous) SIS with $p = \infty$.

Proof.

Suppose $\mathbf{x} \neq \mathbf{x}'$ are a collision. Then

$$\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' \bmod q \iff \mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0} \bmod q, \quad \text{where} \quad \|\mathbf{x} - \mathbf{x}'\|_\infty \leq B.$$

Cryptographic application: collision-resistant hashing.

Consider a family of functions $f_{\mathbf{A}} \colon \{0, 1, \ldots, B\}^m \to \mathbb{Z}_q^n$ given by

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q, \quad \text{where} \quad \mathbf{A} \in \mathbb{Z}_q^{n \times m}.$$

- Let $m \cdot \log(B+1) > n \log q$, so that $f_{\mathbf{A}}$ is *compressing*.
- By sampling $\mathbf{A}$ uniformly at random, $f_{\mathbf{A}}$ becomes a *keyed hash function*.

Lemma

$f_{\mathbf{A}}$ is collision-resistance under (homogeneous) SIS with $p = \infty$.

Proof.

Suppose $\mathbf{x} \neq \mathbf{x}'$ are a collision. Then

$$\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' \bmod q \iff \mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0} \bmod q, \quad \text{where} \quad \|\mathbf{x} - \mathbf{x}'\|_\infty \leq B.$$

$\Rightarrow \mathbf{x} - \mathbf{x}'$ is an $\mathrm{SIS}_{B,\infty}$ solution. $\square$

Learning With Errors

Learning With Errors

- Let n, q be positive integers,

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

Learning With Errors

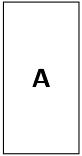
- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

Sample $L_{\mathbf{s},\chi_e,q}$

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

$$\mathbf{A} = \begin{bmatrix} \text{——} \ \mathbf{a}_1 \ \text{——} \\ \vdots \\ \text{——} \ \mathbf{a}_m \ \text{——} \end{bmatrix}$$



Sample $L_{\mathbf{s},\chi_e,q}$

$\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n$,

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

Sample $L_{\mathbf{s},\chi_e,q}$

$$\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n, \quad e_i \xleftarrow{\$} \chi_e$$

$$\mathbf{A} = \begin{bmatrix} \text{---} \ \mathbf{a}_1 \ \text{---} \\ \vdots \\ \text{---} \ \mathbf{a}_m \ \text{---} \end{bmatrix}$$

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

$$\mathbf{A} = \begin{bmatrix} \text{---} & \mathbf{a}_1 & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{a}_m & \text{---} \end{bmatrix}$$

Sample $L_{\mathbf{s},\chi_e,q}$

$\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n, \quad e_i \xleftarrow{\$} \chi_e$

$b_i \leftarrow \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q$

return $(\mathbf{a}_i, b_i)$

$$\boxed{\mathbf{b}} = \boxed{\mathbf{A}}\,\boxed{\mathbf{s}} + \boxed{\mathbf{e}} \quad \bmod q,$$

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

$$\mathbf{A} = \begin{bmatrix} \text{---} \ \mathbf{a}_1 \ \text{---} \\ \vdots \\ \text{---} \ \mathbf{a}_m \ \text{---} \end{bmatrix}$$

Sample $L_{\mathbf{s},\chi_e,q}$

$\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n, \quad e_i \xleftarrow{\$} \chi_e$

$b_i \leftarrow \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q$

return $(\mathbf{a}_i, b_i)$

$$\boxed{\mathbf{b}} = \boxed{\mathbf{A}}\,\boxed{\mathbf{s}} + \boxed{\mathbf{e}} \mod q,$$

Definition

Decision-LWE

Search-LWE

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

$$\mathbf{A} = \begin{bmatrix} \text{---} \ \mathbf{a}_1 \ \text{---} \\ \vdots \\ \text{---} \ \mathbf{a}_m \ \text{---} \end{bmatrix}$$

$$\boxed{\mathbf{b}} = \boxed{\mathbf{A}}\ \boxed{\mathbf{s}} + \boxed{\mathbf{e}} \quad \bmod q,$$

Sample $L_{\mathbf{s},\chi_e,q}$

$\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n, \quad e_i \xleftarrow{\$} \chi_e$

$b_i \leftarrow \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q$

return $(\mathbf{a}_i, b_i)$

Definition

Decision-LWE Guess whether $\{(\mathbf{a}_i, b_i)\}_{i=1}^m \leftarrow L_{\mathbf{s},\chi_e,q}$ or $\{(\mathbf{a}_i, b_i)\}_{i=1}^m \leftarrow U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.

Search-LWE

Learning With Errors

- Let n, q be positive integers,
- χ_e be a probability distribution on $\mathbb{Z}$
- χ_s be a probability distribution on $\mathbb{Z}_q$.
  - Let $\mathbf{s} \leftarrow \chi_s^n$ be a secret vector in $\mathbb{Z}_q^n$.

$$\mathbf{A} = \begin{bmatrix} \text{---} & \mathbf{a}_1 & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{a}_m & \text{---} \end{bmatrix}$$

$$\mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \quad \bmod q,$$

Sample $L_{\mathbf{s},\chi_e,q}$

$\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n, \quad e_i \xleftarrow{\$} \chi_e$

$b_i \leftarrow \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q$

return $(\mathbf{a}_i, b_i)$

Definition

Decision-LWE Guess whether $\{(\mathbf{a}_i, b_i)\}_{i=1}^m \leftarrow L_{\mathbf{s},\chi_e,q}$ or $\{(\mathbf{a}_i, b_i)\}_{i=1}^m \leftarrow U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.

Search-LWE Given $\{(\mathbf{a}_i, b_i)\}_{i=1}^m \leftarrow L_{\mathbf{s},\chi_e,q}$, recover $\mathbf{s}$.

Secret distribution χ_s

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").
- Many proposals: centred binomial, bounded-unfiform $U([-\eta, \eta])$, binary, sparse-binary (!!).

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").
- Many proposals: centred binomial, bounded-unfiform $U([-\eta, \eta])$, binary, sparse-binary (!!).

Error distribution χ_s

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").
- Many proposals: centred binomial, bounded-unfiform $U([-\eta, \eta])$, binary, sparse-binary (!!).

Error distribution χ_s

- χ_e must be *narrow*, to achieve functionality.

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").
- Many proposals: centred binomial, bounded-unfiform $U([-\eta, \eta])$, binary, sparse-binary (!!).

Error distribution χ_s

- χ_e must be *narrow*, to achieve functionality.
- Originally, a *discrete* Gaussian distribution used, to simplify theorem-proving.

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").
- Many proposals: centred binomial, bounded-unfiform $U([-\eta, \eta])$, binary, sparse-binary (!!).

Error distribution χ_s

- χ_e must be *narrow*, to achieve functionality.
- Originally, a *discrete* Gaussian distribution used, to simplify theorem-proving.
- In practice, centred binomial and bounded-uniform distributions are common.

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").
- Many proposals: centred binomial, bounded-unfiform $U([-\eta, \eta])$, binary, sparse-binary (!!).

Error distribution χ_s

- χ_e must be *narrow*, to achieve functionality.
- Originally, a *discrete* Gaussian distribution used, to simplify theorem-proving.
- In practice, centred binomial and bounded-uniform distributions are common.

Definition (Discrete Gaussian distribution)

Let $f_{N(\mu,\sigma^2)}(x)$ be the p.d.f. of the Gaussian distribution.

Secret distribution χ_s

- Original definition: $\chi_s = U(\mathbb{Z}_q)$.
- Standard transformation [7], [8] maps m samples with $\chi_s = U(\mathbb{Z}_q)$ into $m - n$ samples with $\chi_s = \chi_e$ ("normal form LWE").
- Many proposals: centred binomial, bounded-unfiform $U([-\eta, \eta])$, binary, sparse-binary (!!).

Error distribution χ_s

- χ_e must be *narrow*, to achieve functionality.
- Originally, a *discrete* Gaussian distribution used, to simplify theorem-proving.
- In practice, centred binomial and bounded-uniform distributions are common.

Definition (Discrete Gaussian distribution)

Let $f_{N(\mu,\sigma^2)}(x)$ be the p.d.f. of the Gaussian distribution.The *discrete* Gaussian has p.m.f.

$$f_{D_{\mu,\sigma}} : \mathbb{Z} \to [0,1], \quad \text{where} \quad f_{D_{\mu,\sigma}}(x) := \frac{f_{N(\mu,\sigma^2)}(x)}{f_{N(\mu,\sigma^2)}(\mathbb{Z})} = \frac{f_{N(\mu,\sigma^2)}(x)}{\sum_{y\in\mathbb{Z}} f_{N(\mu,\sigma^2)}(y)}.$$

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!
- This allows us to build IND-secure primitives easily.

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!
- This allows us to build IND-secure primitives easily.
- We'll sketch how the equivalence works.
  - For simplicity, we will assume prime q from now on.

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!
- This allows us to build IND-secure primitives easily.
- We'll sketch how the equivalence works.
  - For simplicity, we will assume prime q from now on.

Lemma (Decision-LWE hard $\Rightarrow$ Search-LWE hard)

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!
- This allows us to build IND-secure primitives easily.
- We'll sketch how the equivalence works.
  - For simplicity, we will assume prime q from now on.

Lemma (Decision-LWE hard $\Rightarrow$ Search-LWE hard)

- *Suppose you can solve Search-LWE.*

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!
- This allows us to build IND-secure primitives easily.
- We'll sketch how the equivalence works.
  - For simplicity, we will assume prime q from now on.

Lemma (Decision-LWE hard $\Rightarrow$ Search-LWE hard)

- *Suppose you can solve Search-LWE.*
- *You are given a Decision-LWE challenge:* $(\mathbf{A}, \mathbf{b})$

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!
- This allows us to build IND-secure primitives easily.
- We'll sketch how the equivalence works.
  - For simplicity, we will assume prime q from now on.

Lemma (Decision-LWE hard $\Rightarrow$ Search-LWE hard)

- *Suppose you can solve Search-LWE.*
- *You are given a Decision-LWE challenge:* $(\mathbf{A}, \mathbf{b})$
- *Use the Search-LWE solver to attempt to recover* $\mathbf{s}$ *(and hence* $\mathbf{e}$*).*
  - *If you recover* $\mathbf{s}$ *and a short* $\mathbf{e}$*, guess "LWE".*
  - *Else guess "uniform".*

Search LWE $\iff$ Decision-LWE

- The Search and the Decision variants of LWE enjoy polynomial equivalence!
- This allows us to build IND-secure primitives easily.
- We'll sketch how the equivalence works.
  - For simplicity, we will assume prime q from now on.

Lemma (Decision-LWE hard $\Rightarrow$ Search-LWE hard)

- *Suppose you can solve Search-LWE.*
- *You are given a Decision-LWE challenge:* $(\mathbf{A}, \mathbf{b})$
- *Use the Search-LWE solver to attempt to recover* $\mathbf{s}$ *(and hence* $\mathbf{e}$*).*
  - *If you recover* $\mathbf{s}$ *and a short* $\mathbf{e}$*, guess "LWE".*
  - *Else guess "uniform".*
- *Works because for a random, a short* $\mathbf{e}$ *satisfying the equation is very unlikely to exist.*

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$*.* *Suppose you can solve Decision-LWE.*

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$.

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$, *noting:*

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$, *noting:*

$$\mathbf{b}' = \mathbf{A}\mathbf{s} + \mathbf{e} + z_1\mathbf{u}$$

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$, *noting:*

$$\mathbf{b}' = \mathbf{A}\mathbf{s} + \mathbf{e} + z_1\mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{e} + s_1\mathbf{u} + \Delta\mathbf{u}$$

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$, *noting:*

$$\mathbf{b}' = \mathbf{A}\mathbf{s} + \mathbf{e} + z_1\mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{e} + s_1\mathbf{u} + \Delta\mathbf{u} = [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]\mathbf{s} + \mathbf{e} + \Delta\mathbf{u}$$

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$*. Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$*, let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$*.* $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$*, and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$*, noting:*

$$\mathbf{b}' = \mathbf{A}\mathbf{s} + \mathbf{e} + z_1\mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{e} + s_1\mathbf{u} + \Delta\mathbf{u} = [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]\mathbf{s} + \mathbf{e} + \Delta\mathbf{u} = \mathbf{A}'\mathbf{s} + \mathbf{e} + \Delta\mathbf{u}$$

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$, *noting:*

$$\mathbf{b}' = \mathbf{A}\mathbf{s} + \mathbf{e} + z_1\mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{e} + s_1\mathbf{u} + \Delta\mathbf{u} = [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]\mathbf{s} + \mathbf{e} + \Delta\mathbf{u} = \mathbf{A}'\mathbf{s} + \mathbf{e} + \Delta\mathbf{u}$$

- $\Rightarrow (\mathbf{A}', \mathbf{b}')$ *are*
  - *LWE samples if* $\Delta = 0 \Rightarrow z_1 = s_1$,
  - *Uniformly random otherwise (by* $\mathbf{u}$ *"padding"* $\mathbf{b}$*).*

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$, *noting:*

$$\mathbf{b}' = \mathbf{A}\mathbf{s} + \mathbf{e} + z_1\mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{e} + s_1\mathbf{u} + \Delta\mathbf{u} = [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]\mathbf{s} + \mathbf{e} + \Delta\mathbf{u} = \mathbf{A}'\mathbf{s} + \mathbf{e} + \Delta\mathbf{u}$$

- $\Rightarrow (\mathbf{A}', \mathbf{b}')$ *are*
  - *LWE samples if* $\Delta = 0 \Rightarrow z_1 = s_1$,
  - *Uniformly random otherwise (by* $\mathbf{u}$ *"padding"* $\mathbf{b}$*).*
- *Then the Decision-LWE solver will tell you "LWE" iff* $s_1 = z_1$.

Lemma (Search-LWE hard $\Rightarrow$ Decision-LWE hard)

- *WLOG, we focus on recovering* $\mathbf{s}_1$. *Suppose you can solve Decision-LWE.*
- *Given* $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q)$, *let* $\mathbf{A} = [\mathbf{A}_1 \mid \mathbf{A}_{\dots}]$.
- *Let* $z_1 \in \mathbb{Z}_q$ *be a guess for the value of* $\mathbf{s}_1$. $z_1 = \mathbf{s}_1 + \Delta$ *with* $\Delta = 0$ *iff the guess is correct.*
- *Sample* $\mathbf{u} \sim U(\mathbb{Z}_q^m)$, *and define:* $\mathbf{b}' := \mathbf{b} + z_1\mathbf{u}$, $\mathbf{A}' := [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]$.
- *Call the Decision-LWE solver over* $(\mathbf{A}', \mathbf{b}')$, *noting:*

$$\mathbf{b}' = \mathbf{A}\mathbf{s} + \mathbf{e} + z_1\mathbf{u} = \mathbf{A}\mathbf{s} + \mathbf{e} + s_1\mathbf{u} + \Delta\mathbf{u} = [\mathbf{A}_1 + \mathbf{u} \mid \mathbf{A}_{\dots}]\mathbf{s} + \mathbf{e} + \Delta\mathbf{u} = \mathbf{A}'\mathbf{s} + \mathbf{e} + \Delta\mathbf{u}$$

- $\Rightarrow (\mathbf{A}', \mathbf{b}')$ *are*
  - *LWE samples if* $\Delta = 0 \Rightarrow z_1 = s_1$,
  - *Uniformly random otherwise (by* $\mathbf{u}$ *"padding"* $\mathbf{b}$*).*
- *Then the Decision-LWE solver will tell you "LWE" iff* $s_1 = z_1$.
- *By repeating this* $O(q \cdot n)$ *times, we recover* $\mathbf{s}$.

Inhomogeneous-SIS (I-SIS) with "planted" solutions as LWE

Inhomogeneous-SIS (I-SIS) with “planted” solutions as LWE

- LWE can be seen as a variant of I-SIS, where an unusually short solution was “planted”.

Inhomogeneous-SIS (I-SIS) with "planted" solutions as LWE

- LWE can be seen as a variant of I-SIS, where an unusually short solution was "planted".
- For any SIS instance, we can use the Gaussian heuristic to estimate the norm of solutions.

Inhomogeneous-SIS (I-SIS) with “planted” solutions as LWE

- LWE can be seen as a variant of I-SIS, where an unusually short solution was “planted”.
- For any SIS instance, we can use the Gaussian heuristic to estimate the norm of solutions.

$$\boxed{\mathbf{A}_0}\boxed{\quad\mathbf{A}_1\quad}$$

I-SIS with “planted” solution as LWE.

- Let $\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{A}_1] \leftarrow U(\mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^{n\times(m-n)})$, with high probability, $\det(\mathbf{A}_0) \neq 0 \bmod q$.

Inhomogeneous-SIS (I-SIS) with "planted" solutions as LWE

- LWE can be seen as a variant of I-SIS, where an unusually short solution was "planted".
- For any SIS instance, we can use the Gaussian heuristic to estimate the norm of solutions.

$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{y} \end{bmatrix}$$

I-SIS with "planted" solution as LWE.

- Let $\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{A}_1] \leftarrow U(\mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^{n\times(m-n)})$, with high probability, $\det(\mathbf{A}_0) \neq 0 \bmod q$.
- Pick an usnusually short $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{Z}^n \times \mathbb{Z}^{m-n}$ and let $\mathbf{y} := \mathbf{A}\mathbf{x} \bmod q$.

Inhomogeneous-SIS (I-SIS) with "planted" solutions as LWE

- LWE can be seen as a variant of I-SIS, where an unusually short solution was "planted".
- For any SIS instance, we can use the Gaussian heuristic to estimate the norm of solutions.

$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{y} \end{bmatrix} \Leftrightarrow \begin{bmatrix} \mathbf{I}_n & \mathbf{A}_0^{-1} \times \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix}$$

I-SIS with "planted" solution as LWE.

- Let $\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{A}_1] \leftarrow U(\mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^{n\times(m-n)})$, with high probability, $\det(\mathbf{A}_0) \neq 0 \bmod q$.
- Pick an usnusually short $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{Z}^n \times \mathbb{Z}^{m-n}$ and let $\mathbf{y} := \mathbf{A}\mathbf{x} \bmod q$.
- Multiply through by $\mathbf{A}_0^{-1} \bmod q$.

Inhomogeneous-SIS (I-SIS) with “planted” solutions as LWE

- LWE can be seen as a variant of I-SIS, where an unusually short solution was “planted”.
- For any SIS instance, we can use the Gaussian heuristic to estimate the norm of solutions.

$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} = \mathbf{y} \Leftrightarrow \begin{bmatrix} \mathbf{I}_n & \mathbf{A}_0^{-1} \times \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} = \mathbf{x}_0 + \widetilde{\mathbf{A}} \, \mathbf{x}_1 = \widetilde{\mathbf{y}}$$

I-SIS with “planted” solution as LWE.

- Let $\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{A}_1] \leftarrow U(\mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^{n\times(m-n)})$, with high probability, $\det(\mathbf{A}_0) \neq 0 \bmod q$.
- Pick an usnusually short $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{Z}^n \times \mathbb{Z}^{m-n}$ and let $\mathbf{y} := \mathbf{A}\mathbf{x} \bmod q$.
- Multiply through by $\mathbf{A}_0^{-1} \bmod q$. Let $\widetilde{\mathbf{A}} = \mathbf{A}_0^{-1} \times \mathbf{A}_1$ and $\widetilde{\mathbf{y}} = \mathbf{A}_0^{-1}\mathbf{y}$.

Inhomogeneous-SIS (I-SIS) with "planted" solutions as LWE

- LWE can be seen as a variant of I-SIS, where an unusually short solution was "planted".
- For any SIS instance, we can use the Gaussian heuristic to estimate the norm of solutions.

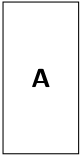
$$\begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} = \mathbf{y} \Leftrightarrow \begin{bmatrix} \mathbf{I}_n & \mathbf{A}_0^{-1} \times \mathbf{A}_1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} = \mathbf{x}_0 + \widetilde{\mathbf{A}} \, \mathbf{x}_1 = \widetilde{\mathbf{y}}$$

I-SIS with "planted" solution as LWE.

- Let $\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{A}_1] \leftarrow U(\mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^{n\times(m-n)})$, with high probability, $\det(\mathbf{A}_0) \neq 0 \bmod q$.
- Pick an usnusually short $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{Z}^n \times \mathbb{Z}^{m-n}$ and let $\mathbf{y} := \mathbf{A}\mathbf{x} \bmod q$.
- Multiply through by $\mathbf{A}_0^{-1} \bmod q$. Let $\widetilde{\mathbf{A}} = \mathbf{A}_0^{-1} \times \mathbf{A}_1$ and $\widetilde{\mathbf{y}} = \mathbf{A}_0^{-1}\mathbf{y}$.
- $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{y}} = \mathbf{A}_0^{-1}\mathbf{y})$ are n LWE samples with secret $\mathbf{x}_1 \in \mathbb{Z}^{m-n}$, and error vector $\mathbf{x}_0$.

LWE as I-SIS with "planted" solutions

LWE as I-SIS with “planted” solutions



LWE as I-SIS with “planted” solution.

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be an LWE instance with secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error $\mathbf{e} \in \mathbb{Z}^m$.

LWE as I-SIS with “planted” solutions

$$\boxed{\mathbf{A}^{\perp}}\ \boxed{\mathbf{A}} = \boxed{\mathbf{0}} \mod q$$

LWE as I-SIS with “planted” solution.

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ be an LWE instance with secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error $\mathbf{e} \in \mathbb{Z}^m$.
- Let $\mathbf{A}^{\perp} \in \mathbb{Z}_q^{n\times m}$ be a matrix where the rows form a set of n independent vectors in the left-kernel of $\mathbf{A}$.

LWE as I-SIS with “planted” solutions

$$\boxed{\mathbf{A}^{\perp}}\ \boxed{\mathbf{A}} = \boxed{\mathbf{0}} \mod q$$

LWE as I-SIS with “planted” solution.

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be an LWE instance with secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error $\mathbf{e} \in \mathbb{Z}^m$.
- Let $\mathbf{A}^{\perp} \in \mathbb{Z}_q^{n \times m}$ be a matrix where the rows form a set of n independent vectors in the left-kernel of $\mathbf{A}$.
- Then

$$\mathbf{A}^{\perp}\mathbf{b} = \mathbf{A}^{\perp}(\mathbf{A}\mathbf{s} + \mathbf{e})$$

LWE as I-SIS with "planted" solutions

$$\boxed{\mathbf{A}^{\perp}} \; \boxed{\mathbf{A}} = \boxed{\mathbf{0}} \mod q$$

LWE as I-SIS with "planted" solution.

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be an LWE instance with secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error $\mathbf{e} \in \mathbb{Z}^m$.
- Let $\mathbf{A}^{\perp} \in \mathbb{Z}_q^{n \times m}$ be a matrix where the rows form a set of n independent vectors in the left-kernel of $\mathbf{A}$.
- Then

$$\mathbf{A}^{\perp}\mathbf{b} = \mathbf{A}^{\perp}(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{\mathbf{A}^{\perp}\mathbf{A}\mathbf{s}} + \mathbf{A}^{\perp}\mathbf{e}$$

LWE as I-SIS with “planted” solutions

$$\boxed{\mathbf{A}^{\perp}}\ \boxed{\mathbf{A}} = \boxed{\mathbf{0}} \mod q$$

LWE as I-SIS with “planted” solution.

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be an LWE instance with secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error $\mathbf{e} \in \mathbb{Z}^m$.
- Let $\mathbf{A}^{\perp} \in \mathbb{Z}_q^{n \times m}$ be a matrix where the rows form a set of n independent vectors in the left-kernel of $\mathbf{A}$.
- Then

$$\mathbf{A}^{\perp}\mathbf{b} = \mathbf{A}^{\perp}(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{\mathbf{A}^{\perp}\mathbf{A}\mathbf{s}} + \mathbf{A}^{\perp}\mathbf{e} = \mathbf{A}^{\perp}\mathbf{e} \bmod q$$

LWE as I-SIS with "planted" solutions

$$\boxed{\mathbf{A}^\perp}\ \boxed{\mathbf{A}} = \boxed{\mathbf{0}} \mod q$$

LWE as I-SIS with "planted" solution.

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ be an LWE instance with secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error $\mathbf{e} \in \mathbb{Z}^m$.
- Let $\mathbf{A}^\perp \in \mathbb{Z}_q^{n\times m}$ be a matrix where the rows form a set of n independent vectors in the left-kernel of $\mathbf{A}$.
- Then

$$\mathbf{A}^\perp\mathbf{b} = \mathbf{A}^\perp(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{\mathbf{A}^\perp\mathbf{A}}\mathbf{s} + \mathbf{A}^\perp\mathbf{e} = \mathbf{A}^\perp\mathbf{e} \bmod q$$

$$\implies \mathbf{e} \text{ is a solution to the I-SIS instance } \mathbf{A}^\perp\mathbf{x} = \mathbf{y} \bmod q \quad \text{where} \quad \mathbf{y} := \mathbf{A}^\perp\mathbf{b} \sim U(\mathbb{Z}_q^n)$$

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

$\mathsf{Enc}(\mathbf{s}, m)$
$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$
$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$
$c \leftarrow (\mathbf{a}, b)$
return $(\mathbf{a}, b)$

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0,1\}$.

$\mathsf{Enc}(\mathbf{s}, m)$

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

$\mathsf{Dec}(\mathbf{s}, c)$

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod {}^{(\pm)} q}{q/2} \right\rceil$

return m'

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

$$
\begin{array}{|l|}
\hline
\mathrm{Enc}(\mathbf{s}, m) \\
\hline
\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e \\
b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q \\
c \leftarrow (\mathbf{a}, b) \\
\textbf{return } (\mathbf{a}, b) \\
\hline
\end{array}
$$

$$
\begin{array}{|l|}
\hline
\mathrm{Dec}(\mathbf{s}, c) \\
\hline
(\mathbf{a}, b) \leftarrow c \\
m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod^{(\pm)} q}{q/2} \right\rceil \\
\textbf{return } m' \\
\hline
\end{array}
$$

Correctness

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

Enc($\mathbf{s}$, m)

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

Dec($\mathbf{s}$, c)

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod {}^{(\pm)} q}{q/2} \right\rceil$

return m'

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

$$
\boxed{\begin{array}{l}
\mathsf{Enc}(\mathbf{s}, m) \\
\hline
\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e \\
b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q \\
c \leftarrow (\mathbf{a}, b) \\
\textbf{return } (\mathbf{a}, b)
\end{array}}
$$

$$
\boxed{\begin{array}{l}
\mathsf{Dec}(\mathbf{s}, c) \\
\hline
(\mathbf{a}, b) \leftarrow c \\
m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod^{(\pm)} q}{q/2} \right\rceil \\
\textbf{return } m'
\end{array}}
$$

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.
- Suppose $e \approx 0$.

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

Enc(s, m)

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

Dec(s, c)

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod {}^{(\pm)} q}{q/2} \right\rceil$

return m'

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.
- Suppose $e \approx 0$.
  - If $m = 0$, $m + e \approx 0 \implies \lfloor \frac{m+e}{q/2} \rceil = 0$.

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

$$
\begin{array}{l}
\mathrm{Enc}(\mathbf{s}, m) \\
\hline
\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e \\
b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q \\
c \leftarrow (\mathbf{a}, b) \\
\textbf{return } (\mathbf{a}, b)
\end{array}
$$

$$
\begin{array}{l}
\mathrm{Dec}(\mathbf{s}, c) \\
\hline
(\mathbf{a}, b) \leftarrow c \\
m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod {}^{(\pm)} q}{q/2} \right\rceil \\
\textbf{return } m'
\end{array}
$$

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.
- Suppose $e \approx 0$.
  - If $m = 0$, $m + e \approx 0 \implies \lfloor \frac{m+e}{q/2} \rceil = 0$.
  - If $m = 1$, $m + e \approx q/2 \implies \lfloor \frac{m+e}{q/2} \rceil = 1$.

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

$\mathrm{Enc}(\mathbf{s}, m)$

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

$\mathrm{Dec}(\mathbf{s}, c)$

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod^{(\pm)} q}{q/2} \right\rceil$

return m'

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.
- Suppose $e \approx 0$.
  - If $m = 0$, $m + e \approx 0 \implies \lfloor \frac{m+e}{q/2} \rceil = 0$.
  - If $m = 1$, $m + e \approx q/2 \implies \lfloor \frac{m+e}{q/2} \rceil = 1$.
- For this to work, we need $|e| < q/4$.

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

Enc$(\mathbf{s}, m)$

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

Dec$(\mathbf{s}, c)$

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod {}^{(\pm)} q}{q/2} \right\rceil$

return m'

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rceil$.
- Suppose $e \approx 0$.
  - If $m = 0$, $m + e \approx 0 \implies \lfloor \frac{m+e}{q/2} \rceil = 0$.
  - If $m = 1$,
    $m + e \approx q/2 \implies \lfloor \frac{m+e}{q/2} \rceil = 1$.
- For this to work, we need $|e| < q/4$.

Security

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

Enc($\mathbf{s}$, m)

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

Dec($\mathbf{s}$, c)

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod {}^{(\pm)} q}{q/2} \right\rceil$

return m'

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.
- Suppose $e \approx 0$.
  - If $m = 0$, $m + e \approx 0 \implies \lfloor \frac{m+e}{q/2} \rceil = 0$.
  - If $m = 1$, $m + e \approx q/2 \implies \lfloor \frac{m+e}{q/2} \rceil = 1$.
- For this to work, we need $|e| < q/4$.

Security

- By Decision-LWE, $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q) \overset{c}{\approx} U(\mathbb{Z}_q^{1 \times n} \times \mathbb{Z}_q)$

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

$\mathsf{Enc}(\mathbf{s}, m)$

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

$\mathsf{Dec}(\mathbf{s}, c)$

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod^{(\pm)} q}{q/2} \right\rceil$

return m'

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.
- Suppose $e \approx 0$.
  - If $m = 0$, $m + e \approx 0 \implies \lfloor \frac{m+e}{q/2} \rceil = 0$.
  - If $m = 1$, $m + e \approx q/2 \implies \lfloor \frac{m+e}{q/2} \rceil = 1$.
- For this to work, we need $|e| < q/4$.

Security

- By Decision-LWE, $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q) \stackrel{c}{\approx} U(\mathbb{Z}_q^{1 \times n} \times \mathbb{Z}_q)$
- This turns b into a "computational one-time pad" encryption of m.

Cryptographic application: IND-CPA symmetric encryption.

- From LWE we can build SE for bits, $m \in \{0, 1\}$.

Enc($\mathbf{s}$, m)

$\mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \quad e \leftarrow \chi_e$

$b \leftarrow \langle \mathbf{a}, \mathbf{s} \rangle + e + m \cdot \lfloor q/2 \rfloor \bmod q$

$c \leftarrow (\mathbf{a}, b)$

return $(\mathbf{a}, b)$

Dec($\mathbf{s}$, c)

$(\mathbf{a}, b) \leftarrow c$

$m' \leftarrow \left\lfloor \dfrac{c - \langle \mathbf{a}, \mathbf{s} \rangle \bmod^{(\pm)} q}{q/2} \right\rceil$

return m'

Correctness

- $c - \langle \mathbf{a}, \mathbf{s} \rangle = e + m \cdot \lfloor q/2 \rfloor$.
- Suppose $e \approx 0$.
  - If $m = 0$, $m + e \approx 0 \implies \lfloor \frac{m+e}{q/2} \rceil = 0$.
  - If $m = 1$,
    $m + e \approx q/2 \implies \lfloor \frac{m+e}{q/2} \rceil = 1$.
- For this to work, we need $|e| < q/4$.

Security

- By Decision-LWE,
  $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q) \overset{c}{\approx} U(\mathbb{Z}_q^{1 \times n} \times \mathbb{Z}_q)$
- This turns b into a "computational one-time pad" encryption of m.
- IND-CPA follows from each ciphertext using its own pad.

On the similarities of LWE, DLOG and DH.

- When designing new primitives, it can be useful to imitate prior constructions.

On the similarities of LWE, DLOG and DH.

- When designing new primitives, it can be useful to imitate prior constructions.
- Especially when there is some urgency for practical, secure constructions.

On the similarities of LWE, DLOG and DH.

- When designing new primitives, it can be useful to imitate prior constructions.
- Especially when there is some urgency for practical, secure constructions.
- Lattice assumptions share some similarities with group-based ones.

On the similarities of LWE, DLOG and DH.

- When designing new primitives, it can be useful to imitate prior constructions.
- Especially when there is some urgency for practical, secure constructions.
- Lattice assumptions share some similarities with group-based ones.

Similarities with DLOG

- In pre-quantum cryptography, we relied on the DLOG problem:
  - Given $(g, h = g^x) \in \mathbb{G} \times \mathbb{G}$, recover x.

On the similarities of LWE, DLOG and DH.

- When designing new primitives, it can be useful to imitate prior constructions.
- Especially when there is some urgency for practical, secure constructions.
- Lattice assumptions share some similarities with group-based ones.

Similarities with DLOG

- In pre-quantum cryptography, we relied on the DLOG problem:
  - Given $(g, h = g^x) \in \mathbb{G} \times \mathbb{G}$, recover x.
- This is not too different from
  - Given $\mathbf{Ax} = \mathbf{b}$ recover $\mathbf{x}$ (SIS)

On the similarities of LWE, DLOG and DH.

- When designing new primitives, it can be useful to imitate prior constructions.
- Especially when there is some urgency for practical, secure constructions.
- Lattice assumptions share some similarities with group-based ones.

Similarities with DLOG

- In pre-quantum cryptography, we relied on the DLOG problem:
  - Given $(g, h = g^x) \in \mathbb{G} \times \mathbb{G}$, recover x.
- This is not too different from
  - Given $\mathbf{Ax} = \mathbf{b}$ recover $\mathbf{x}$ (SIS)
  - Given $\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e}$ recover $\mathbf{s}$ (LWE)

Similarities with DDH

- To build in cryptomania, we needed extra functionality, such as DDH:
  - Distinguish (g, g^x, g^y, g^{xy}) from (g, g^x, g^y, g^z), where $g^x, g^y, g^z \overset{\text{iid}}{\sim} U(\mathbb{G})$.

Similarities with DDH

- To build in cryptomania, we needed extra functionality, such as DDH:
  - Distinguish (g, g^x, g^y, g^{xy}) from (g, g^x, g^y, g^z), where $g^x, g^y, g^z \overset{\text{iid}}{\sim} U(\mathbb{G})$.
- This is not too different from:
  - Given $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$, distinguish $(\mathbf{A},\ \mathbf{As} + \mathbf{e},\ \mathbf{z}^T\mathbf{A} + \mathbf{f},\ \mathbf{z}^T(\mathbf{As} + \mathbf{e}) + e')$ from uniform.

Similarities with DDH

- To build in cryptomania, we needed extra functionality, such as DDH:
  - Distinguish (g, g^x, g^y, g^{xy}) from (g, g^x, g^y, g^z), where $g^x, g^y, g^z \overset{\text{iid}}{\sim} U(\mathbb{G})$.
- This is not too different from:
  - Given $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$, distinguish $(\mathbf{A},\ \mathbf{As} + \mathbf{e},\ \mathbf{z}^T\mathbf{A} + \mathbf{f},\ \mathbf{z}^T(\mathbf{As} + \mathbf{e}) + e')$ from uniform.
  - And this is hard if Decision-LWE is hard for $\geq n + 1$ samples!

Similarities with DDH

- To build in cryptomania, we needed extra functionality, such as DDH:
  - Distinguish (g, g^x, g^y, g^{xy}) from (g, g^x, g^y, g^z), where $g^x, g^y, g^z \stackrel{\text{iid}}{\sim} U(\mathbb{G})$.
- This is not too different from:
  - Given $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$, distinguish $(\mathbf{A},\ \mathbf{A}\mathbf{s}+\mathbf{e},\ \mathbf{z}^T\mathbf{A}+\mathbf{f},\ \mathbf{z}^T(\mathbf{A}\mathbf{s}+\mathbf{e})+e')$ from uniform.
  - And this is hard if Decision-LWE is hard for $\geq n+1$ samples!

Proof.

$$
\begin{array}{rcccc}
 & (\mathbf{A}, & \mathbf{A}\mathbf{s}+\mathbf{e}, & \mathbf{z}^T\mathbf{A}+\mathbf{f}, & \mathbf{z}^T(\mathbf{A}\mathbf{s}+\mathbf{e})+e') \\
\stackrel{c}{\approx} & (\mathbf{A}, & \mathbf{u}, & \mathbf{z}^T\mathbf{A}+\mathbf{f}, & \mathbf{z}^T\mathbf{u}+e') \\
\stackrel{c}{\approx} & (\mathbf{A}, & \mathbf{u}, & \mathbf{w}, & \mathbf{z}^T\mathbf{u}+e') \\
\stackrel{c}{\approx} & (\mathbf{A}, & \mathbf{u}, & \mathbf{w}, & w'),
\end{array}
$$

for $\mathbf{u}_i, \mathbf{w}_i, w' \stackrel{\text{iid}}{\sim} U(\mathbb{Z}_q)$. □

Why SIS and LWE are “lattice-based”: the provable angle

Why SIS and LWE are “lattice-based”: the provable angle

- There exist polynomial-time (in n) reductions from worst-case lattice problems (eg., SIVP) to average-case SIS/LWE.

Why SIS and LWE are "lattice-based": the provable angle

- There exist polynomial-time (in n) reductions from worst-case lattice problems (eg., SIVP) to average-case SIS/LWE.
- This means that cryptanalytic attacks against SIS/LWE imply solving lattice problems *for all lattices*, usually a in lower dimension,

Why SIS and LWE are “lattice-based”: the provable angle

- There exist polynomial-time (in n) reductions from worst-case lattice problems (eg., SIVP) to average-case SIS/LWE.
- This means that cryptanalytic attacks against SIS/LWE imply solving lattice problems *for all lattices*, usually a in lower dimension,eg.
  - From worst-case $\text{SIVP}_{\tilde{O}(n)}$ to average-case SIS with $p = \infty$, $B = 1$, $q \gg m \approx n \log q$ [9].

Why SIS and LWE are “lattice-based”: the provable angle

- There exist polynomial-time (in n) reductions from worst-case lattice problems (eg., SIVP) to average-case SIS/LWE.
- This means that cryptanalytic attacks against SIS/LWE imply solving lattice problems *for all lattices*, usually a in lower dimension,eg.
  - From worst-case $\mathrm{SIVP}_{\tilde{O}(n)}$ to average-case SIS with $p = \infty$, $B = 1$, $q \gg m \approx n \log q$ [9].
  - From worst-case $\mathrm{BDD}_{n^{-1/2}}$ to average-case LWE with $m \in \mathrm{poly}(n)$, χ_e discrete Gaussian with $\sigma/q \in (0, 1)$ [10].

Solving SIS and LWE

Fernando Virdia — `https://fundamental.domains`
EPFL-ETH Summer School on Lattice-based Cryptography, July 2025

Why SIS and LWE are "lattice-based": the practical angle

- We've mentioned polynomial-time reductions between worst-case lattice problems, and SIS/LWE.

Why SIS and LWE are "lattice-based": the practical angle

- We've mentioned polynomial-time reductions between worst-case lattice problems, and SIS/LWE.
- As a theoretician, this may put you at ease about lattice-based cryptography.

Why SIS and LWE are "lattice-based": the practical angle

- We've mentioned polynomial-time reductions between worst-case lattice problems, and SIS/LWE.
- As a theoretician, this may put you at ease about lattice-based cryptography.
- However, we don't usually actually choose SIS/LWE parameters based on reductions!
  - In practice, these can be extremely practically untight [11, § 6]

Why SIS and LWE are "lattice-based": the practical angle

- We've mentioned polynomial-time reductions between worst-case lattice problems, and SIS/LWE.
- As a theoretician, this may put you at ease about lattice-based cryptography.
- However, we don't usually actually choose SIS/LWE parameters based on reductions!
  - In practice, these can be extremely practically untight [11, § 6]
- Instead, we investigate attacks on SIS/LWE directly, analyse their runtime, verify this experimentally, and use scripts such as the *lattice-estimator* to extrapolate attack costs.

Why SIS and LWE are "lattice-based": the practical angle

- We've mentioned polynomial-time reductions between worst-case lattice problems, and SIS/LWE.
- As a theoretician, this may put you at ease about lattice-based cryptography.
- However, we don't usually actually choose SIS/LWE parameters based on reductions!
  - In practice, these can be extremely practically untight [11, § 6]
- Instead, we investigate attacks on SIS/LWE directly, analyse their runtime, verify this experimentally, and use scripts such as the *lattice-estimator* to extrapolate attack costs.
- The most practical attacks are based on *lattice reduction*.

Why SIS and LWE are "lattice-based": the practical angle

- We've mentioned polynomial-time reductions between worst-case lattice problems, and SIS/LWE.
- As a theoretician, this may put you at ease about lattice-based cryptography.
- However, we don't usually actually choose SIS/LWE parameters based on reductions!
  - In practice, these can be extremely practically untight [11, § 6]
- Instead, we investigate attacks on SIS/LWE directly, analyse their runtime, verify this experimentally, and use scripts such as the *lattice-estimator* to extrapolate attack costs.
- The most practical attacks are based on *lattice reduction*.
- Internally, these require oracles solving lattice problems, such as approx-SVP.

Why SIS and LWE are "lattice-based": the practical angle

- We've mentioned polynomial-time reductions between worst-case lattice problems, and SIS/LWE.
- As a theoretician, this may put you at ease about lattice-based cryptography.
- However, we don't usually actually choose SIS/LWE parameters based on reductions!
  - In practice, these can be extremely practically untight [11, § 6]
- Instead, we investigate attacks on SIS/LWE directly, analyse their runtime, verify this experimentally, and use scripts such as the *lattice-estimator* to extrapolate attack costs.
- The most practical attacks are based on *lattice reduction*.
- Internally, these require oracles solving lattice problems, such as approx-SVP.
- Now we will talk about such attacks.

Linear algebra

- Given a basis $\mathbf{B}$, we can derive an orthogonal basis $\mathbf{B}^*$ via the Gram–Schmidt process.
- The rows of $\mathbf{B}^*$ are

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j<i} \mu_{i,j} \mathbf{b}_j^* \quad \text{for} \quad i \in [d], \quad \text{where} \quad \mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2 \quad \text{for} \quad i > j.$$

Linear algebra

- Given a basis $\mathbf{B}$, we can derive an orthogonal basis $\mathbf{B}^*$ via the Gram–Schmidt process.
- The rows of $\mathbf{B}^*$ are

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j<i} \mu_{i,j} \mathbf{b}_j^* \quad \text{for} \quad i \in [d], \quad \text{where} \quad \mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2 \quad \text{for} \quad i > j.$$

- In matrix form,

$$\begin{bmatrix} \text{---} & \mathbf{b}_1 & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{b}_d & \text{---} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \mu_{2,1} & 1 & & \\ \vdots & \ddots & \ddots & \\ \mu_{d,1} & \dots & \mu_{d,d-1} & 1 \end{bmatrix} \begin{bmatrix} \text{---} & \mathbf{b}_1^* & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{b}_d^* & \text{---} \end{bmatrix}.$$

Linear algebra

- Given a basis $\mathbf{B}$, we can derive an orthogonal basis $\mathbf{B}^*$ via the Gram–Schmidt process.
- The rows of $\mathbf{B}^*$ are

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j<i} \mu_{i,j}\mathbf{b}_j^* \quad \text{for} \quad i \in [d], \quad \text{where} \quad \mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2 \quad \text{for} \quad i > j.$$

- In matrix form,

$$\begin{bmatrix} — & \mathbf{b}_1 & — \\ & \vdots & \\ — & \mathbf{b}_d & — \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \mu_{2,1} & 1 & & \\ \vdots & \ddots & \ddots & \\ \mu_{d,1} & \dots & \mu_{d,d-1} & 1 \end{bmatrix} \begin{bmatrix} — & \mathbf{b}_1^* & — \\ & \vdots & \\ — & \mathbf{b}_d^* & — \end{bmatrix}.$$

Remark

Recall that the volume of a lattice $\Lambda(\mathbf{B})$ is $|\det(\mathbf{B})|$. Given the Gram-Schmidt orthogonalisation $\mathbf{b}_1^*, \dots, \mathbf{b}_n^*$ of the basis $\mathbf{B}$, $|\det(\mathbf{B})| = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|$.

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Definition (Basis profile)

Given $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$, let $\mathbf{B}^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_n^*)$ be its Gram-Schmidt orthogonalization. We define $\mathrm{prof}(\mathbf{B}) \coloneqq (\|\mathbf{b}_1^*\|^2, \dots, \|\mathbf{b}_n^*\|^2)$

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Definition (Basis profile)

Given $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, let $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its Gram-Schmidt orthogonalization. We define $\mathrm{prof}(\mathbf{B}) \coloneqq (\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_n^*\|^2)$

- Two common basis quality metrics:

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Definition (Basis profile)

Given $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, let $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its Gram-Schmidt orthogonalization.

$$\text{We define } \operatorname{prof}(\mathbf{B}) \coloneqq (\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_n^*\|^2)$$

- Two common basis quality metrics:
  - $\|\mathbf{b}_1\|$ (the shortest, the better),

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Definition (Basis profile)

Given $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, let $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its Gram-Schmidt orthogonalization.

$$\text{We define } \mathrm{prof}(\mathbf{B}) := (\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_n^*\|^2)$$

- Two common basis quality metrics:
  - $\|\mathbf{b}_1\|$ (the shortest, the better),
  - How orthogonal its vectors are (the more, the better).

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Definition (Basis profile)

Given $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, let $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its Gram-Schmidt orthogonalization.

$$\text{We define } \operatorname{prof}(\mathbf{B}) \coloneqq (\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_n^*\|^2)$$

- Two common basis quality metrics:
  - $\|\mathbf{b}_1\|$ (the shortest, the better),
  - How orthogonal its vectors are (the more, the better).
- The basis profile captures both properties in the case of q-ary lattices:

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Definition (Basis profile)

Given $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, let $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its Gram-Schmidt orthogonalization.

$$\text{We define } \operatorname{prof}(\mathbf{B}) := (\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_n^*\|^2)$$

- Two common basis quality metrics:
  - $\|\mathbf{b}_1\|$ (the shortest, the better),
  - How orthogonal its vectors are (the more, the better).
- The basis profile captures both properties in the case of q-ary lattices:
  - $\|\mathbf{b}_1^*\|^2 = \|\mathbf{b}_1\|^2$ is included in the profile.

Lattice reduction

- Informally, lattice reduction is any algorithmic technique that takes as input a basis of a lattice and finds a basis of better *quality*.
- To better reason about basis quality, we first introduce the notion of *basis profile*.

Definition (Basis profile)

Given $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$, let $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$ be its Gram-Schmidt orthogonalization.

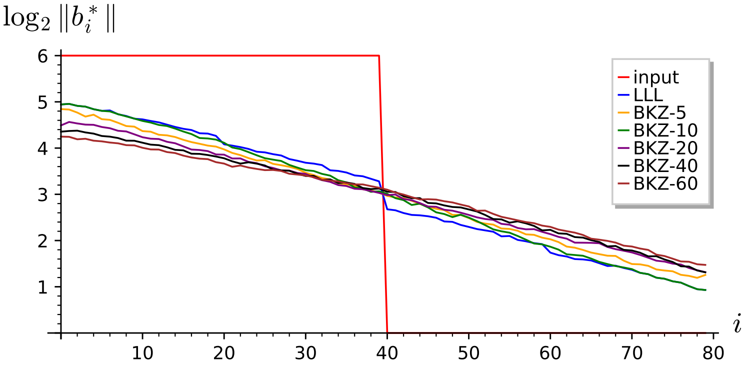
We define $\mathrm{prof}(\mathbf{B}) \coloneqq (\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_n^*\|^2)$

- Two common basis quality metrics:
  - $\|\mathbf{b}_1\|$ (the shortest, the better),
  - How orthogonal its vectors are (the more, the better).
- The basis profile captures both properties in the case of q-ary lattices:
  - $\|\mathbf{b}_1^*\|^2 = \|\mathbf{b}_1\|^2$ is included in the profile.
  - The *flatter* a basis profile is, the closer to orthogonal is the basis.

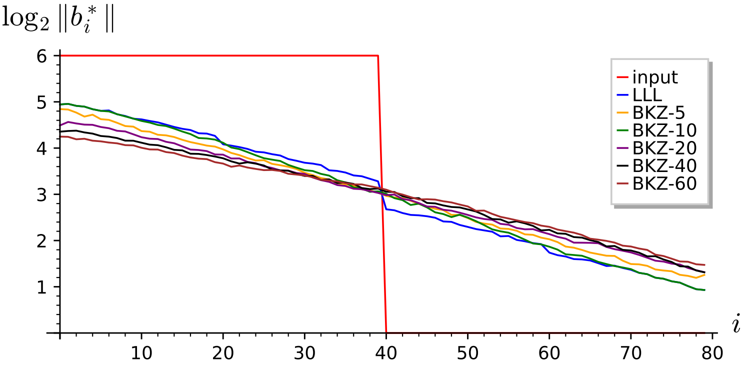
- Lattice reduction algorithms such as LLL and BKZ affect both.

- Lattice reduction algorithms such as LLL and BKZ affect both.
- They output a new basis of better (or non-worse) quality.

- Lattice reduction algorithms such as LLL and BKZ affect both.
- They output a new basis of better (or non-worse) quality.
- In the case of BKZ, the larger its cost parameter β, the flatter the profile, and the shorter the first vector.



- Lattice reduction algorithms such as LLL and BKZ affect both.
- They output a new basis of better (or non-worse) quality.
- In the case of BKZ, the larger its cost parameter β, the flatter the profile, and the shorter the first vector.



Remark

Interestingly, it would appear that the log-plot of a reduced basis profile forms a straight line.

Predicting $\|\mathbf{b}_1^*\|$

- Worst- and average-case guarantees on the output quality of lattice reduction algorithms can be proven.

Predicting $\|\mathbf{b}_1^*\|$

- Worst- and average-case guarantees on the output quality of lattice reduction algorithms can be proven.
- Practical security estimates rely on average-case heuristics on the *Hermite factor*.

Predicting $\|\mathbf{b}_1^*\|$

- Worst- and average-case guarantees on the output quality of lattice reduction algorithms can be proven.
- Practical security estimates rely on average-case heuristics on the *Hermite factor*.

Definition (Hermite factor or Hermite defect [12])

- Let $\mathcal{A}$ be a lattice reduction algorithm and $\mathcal{L}_n = \{\text{rank-}n \text{ lattice basis}\} \equiv GL_n(\mathbb{R})$.

Predicting $\|\mathbf{b}_1^*\|$

- Worst- and average-case guarantees on the output quality of lattice reduction algorithms can be proven.
- Practical security estimates rely on average-case heuristics on the *Hermite factor*.

Definition (Hermite factor or Hermite defect [12])

- Let $\mathcal{A}$ be a lattice reduction algorithm and $\mathcal{L}_n = \{\text{rank-}n \text{ lattice basis}\} \equiv GL_n(\mathbb{R})$.
- The Hermite factor of $\mathcal{A}$ is the real-valued random variable

$$\zeta_{\mathcal{A},n} := \frac{\|\mathbf{b}_1\|}{\mathrm{vol}(\Lambda)^{1/n}} \quad \text{where} \quad \mathbf{b}_1, \ldots, \mathbf{b}_n \xleftarrow{\$} \mathcal{A}(\Lambda \xleftarrow{\$} \mathcal{L}_n),$$

where $\Lambda \xleftarrow{\$} \mathcal{L}_n$ either by working in a finite subset, or by using the Haar measure.

Predicting $\|\mathbf{b}_1^*\|$

- Worst- and average-case guarantees on the output quality of lattice reduction algorithms can be proven.
- Practical security estimates rely on average-case heuristics on the *Hermite factor*.

Definition (Hermite factor or Hermite defect [12])

- Let $\mathcal{A}$ be a lattice reduction algorithm and $\mathcal{L}_n = \{\text{rank-}n \text{ lattice basis}\} \equiv GL_n(\mathbb{R})$.
- The Hermite factor of $\mathcal{A}$ is the real-valued random variable

$$\zeta_{\mathcal{A},n} := \frac{\|\mathbf{b}_1\|}{\mathrm{vol}(\Lambda)^{1/n}} \quad \text{where} \quad \mathbf{b}_1, \ldots, \mathbf{b}_n \xleftarrow{\$} \mathcal{A}(\Lambda \xleftarrow{\$} \mathcal{L}_n),$$

  where $\Lambda \xleftarrow{\$} \mathcal{L}_n$ either by working in a finite subset, or by using the Haar measure.
- $\mathbb{E}[\zeta_{\mathcal{A},n}]$ is the average-case Hermite factor.

Predicting $\|\mathbf{b}_1^*\|$

- Worst- and average-case guarantees on the output quality of lattice reduction algorithms can be proven.
- Practical security estimates rely on average-case heuristics on the *Hermite factor*.

Definition (Hermite factor or Hermite defect [12])

- Let $\mathcal{A}$ be a lattice reduction algorithm and $\mathcal{L}_n = \{\text{rank-}n \text{ lattice basis}\} \equiv GL_n(\mathbb{R})$.
- The Hermite factor of $\mathcal{A}$ is the real-valued random variable

$$\zeta_{\mathcal{A},n} := \frac{\|\mathbf{b}_1\|}{\mathrm{vol}(\Lambda)^{1/n}} \quad \text{where} \quad \mathbf{b}_1, \ldots, \mathbf{b}_n \xleftarrow{\$} \mathcal{A}(\Lambda \xleftarrow{\$} \mathcal{L}_n),$$

  where $\Lambda \xleftarrow{\$} \mathcal{L}_n$ either by working in a finite subset, or by using the Haar measure.
- $\mathbb{E}[\zeta_{\mathcal{A},n}]$ is the average-case Hermite factor.
- We abuse notation and write ζ for $\mathbb{E}[\zeta_{\mathcal{A},n}]$.

Predicting $\|\mathbf{b}_1^*\|$

- Worst- and average-case guarantees on the output quality of lattice reduction algorithms can be proven.
- Practical security estimates rely on average-case heuristics on the *Hermite factor*.

Definition (Hermite factor or Hermite defect [12])

- Let $\mathcal{A}$ be a lattice reduction algorithm and $\mathcal{L}_n = \{\text{rank-}n \text{ lattice basis}\} \equiv GL_n(\mathbb{R})$.
- The Hermite factor of $\mathcal{A}$ is the real-valued random variable

$$\zeta_{\mathcal{A},n} := \frac{\|\mathbf{b}_1\|}{\mathrm{vol}(\Lambda)^{1/n}} \quad \text{where} \quad \mathbf{b}_1, \ldots, \mathbf{b}_n \xleftarrow{\$} \mathcal{A}(\Lambda \xleftarrow{\$} \mathcal{L}_n),$$

  where $\Lambda \xleftarrow{\$} \mathcal{L}_n$ either by working in a finite subset, or by using the Haar measure.
- $\mathbb{E}[\zeta_{\mathcal{A},n}]$ is the average-case Hermite factor.
- We abuse notation and write ζ for $\mathbb{E}[\zeta_{\mathcal{A},n}]$.

- We say an algorithm has Hermite factor ζ if it outputs bases where

$$\|\mathbf{b}_1\| \approx \zeta_n \cdot \mathrm{vol}(\Lambda)^{1/n},$$

with $\approx$ replaced by $\leq$ in worst-case bounds.

- We say an algorithm has Hermite factor ζ if it outputs bases where

$$\|\mathbf{b}_1\| \approx \zeta_n \cdot \mathrm{vol}(\Lambda)^{1/n},$$

with $\approx$ replaced by $\leq$ in worst-case bounds.
- ζ_n can be measured experimentally, or derived analytically.

- We say an algorithm has Hermite factor ζ if it outputs bases where

$$\|\mathbf{b}_1\| \approx \zeta_n \cdot \mathrm{vol}(\Lambda)^{1/n},$$

  with $\approx$ replaced by $\leq$ in worst-case bounds.
- ζ_n can be measured experimentally, or derived analytically.
- For example, LLL with parameter $\delta_{L^3} \in (1/4, 1)$ provably achieves $\zeta_{\mathsf{LLL},n} \leq (\delta_{L^3} - 1/4)^{-(n-1)/4}$ [12, Chap. 2, Thm. 9].

- We say an algorithm has Hermite factor ζ if it outputs bases where

$$\|\mathbf{b}_1\| \approx \zeta_n \cdot \mathrm{vol}(\Lambda)^{1/n},$$

with $\approx$ replaced by $\leq$ in worst-case bounds.
- ζ_n can be measured experimentally, or derived analytically.
- For example, LLL with parameter $\delta_{L^3} \in (1/4, 1)$ provably achieves $\zeta_{\mathsf{LLL},n} \leq (\delta_{L^3} - 1/4)^{-(n-1)/4}$ [12, Chap. 2, Thm. 9].
- For $\delta_{L^3} \to 1$, this $\approx 1.075^{n-1}$, yet experimentally we measure $\zeta_{\mathsf{LLL},n} \approx 1.02^{n-1}$ [13]

- We say an algorithm has Hermite factor ζ if it outputs bases where

$$\|\mathbf{b}_1\| \approx \zeta_n \cdot \mathrm{vol}(\Lambda)^{1/n},$$

with $\approx$ replaced by $\leq$ in worst-case bounds.
- ζ_n can be measured experimentally, or derived analytically.
- For example, LLL with parameter $\delta_{L^3} \in (1/4, 1)$ provably achieves $\zeta_{\mathrm{LLL},n} \leq (\delta_{L^3} - 1/4)^{-(n-1)/4}$ [12, Chap. 2, Thm. 9].
- For $\delta_{L^3} \to 1$, this $\approx 1.075^{n-1}$, yet experimentally we measure $\zeta_{\mathrm{LLL},n} \approx 1.02^{n-1}$ [13]
- For BKZ-β we measure (and can heuristically argue) [2], [14], [15]

$$\zeta_{\mathrm{BKZ}\text{-}\beta,n} \approx \left(\left((\pi\beta)^{1/\beta}\frac{\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}\right)^{n-1}.$$

- We say an algorithm has Hermite factor ζ if it outputs bases where

$$\|\mathbf{b}_1\| \approx \zeta_n \cdot \mathrm{vol}(\Lambda)^{1/n},$$

with $\approx$ replaced by $\leq$ in worst-case bounds.

- ζ_n can be measured experimentally, or derived analytically.
- For example, LLL with parameter $\delta_{L^3} \in (1/4, 1)$ provably achieves $\zeta_{\mathsf{LLL},n} \leq (\delta_{L^3} - 1/4)^{-(n-1)/4}$ [12, Chap. 2, Thm. 9].
- For $\delta_{L^3} \to 1$, this $\approx 1.075^{n-1}$, yet experimentally we measure $\zeta_{\mathsf{LLL},n} \approx 1.02^{n-1}$ [13]
- For BKZ-β we measure (and can heuristically argue) [2], [14], [15]

$$\zeta_{\mathsf{BKZ}\text{-}\beta,n} \approx \left(\left((\pi\beta)^{1/\beta}\frac{\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}\right)^{n-1}.$$

- Due to the exponential scaling, we talk of *root-Hermite factor* $\delta := \zeta^{1/(n-1)}$.

- We say an algorithm has Hermite factor ζ if it outputs bases where

$$\|\mathbf{b}_1\| \approx \zeta_n \cdot \mathrm{vol}(\Lambda)^{1/n},$$

with $\approx$ replaced by $\leq$ in worst-case bounds.
- ζ_n can be measured experimentally, or derived analytically.
- For example, LLL with parameter $\delta_{L^3} \in (1/4, 1)$ provably achieves $\zeta_{\mathrm{LLL},n} \leq (\delta_{L^3} - 1/4)^{-(n-1)/4}$ [12, Chap. 2, Thm. 9].
- For $\delta_{L^3} \to 1$, this $\approx 1.075^{n-1}$, yet experimentally we measure $\zeta_{\mathrm{LLL},n} \approx 1.02^{n-1}$ [13]
- For BKZ-β we measure (and can heuristically argue) [2], [14], [15]

$$\zeta_{\text{BKZ-}\beta,n} \approx \left(\left((\pi\beta)^{1/\beta}\frac{\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}\right)^{n-1}.$$

- Due to the exponential scaling, we talk of *root-Hermite factor* $\delta := \zeta^{1/(n-1)}$.
- LLL has root-Hermite factor $\delta = 1.02$, BKZ-β has $\delta = \left((\pi\beta)^{1/\beta}\frac{\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}$, independent of n.

Predicting $(\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_n^*\|)$

- Beyond $\|\mathbf{b}_1\|$, heuristics also exist about the full profile of reduced bases for random q-ary lattices.

Predicting $(\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_n^*\|)$

- Beyond $\|\mathbf{b}_1\|$, heuristics also exist about the full profile of reduced bases for random q-ary lattices.

Heuristic (Geometric Series Assumption (GSA) [16])

Given a basis $\mathbf{B}$ *output by a lattice reduction algorithm, the norms of the Gram-Schmidt vectors* $\mathbf{b}_i^*$ *satisfy*

$$\|\mathbf{b}_i^*\| = \alpha^{i-1} \cdot \|\mathbf{b}_1\|$$

for some constant $\alpha \in (0, 1)$.

Predicting $(\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_n^*\|)$

- Beyond $\|\mathbf{b}_1\|$, heuristics also exist about the full profile of reduced bases for random q-ary lattices.

Heuristic (Geometric Series Assumption (GSA) [16])

Given a basis $\mathbf{B}$ *output by a lattice reduction algorithm, the norms of the Gram-Schmidt vectors* $\mathbf{b}_i^*$ *satisfy*

$$\|\mathbf{b}_i^*\| = \alpha^{i-1} \cdot \|\mathbf{b}_1\|$$

for some constant $\alpha \in (0, 1)$.

- The GSA captures the straightness of the log-plot of the basis profiles we saw before.
  - $\log \|\mathbf{b}_i^*\| = (i-1) \cdot \log \alpha + \log \|\mathbf{b}_1\|$ is a straight line with slope $\log \alpha$.

A simple computation allows us to deduce the Hermite factor of an algorithm in terms of α.

A simple computation allows us to deduce the Hermite factor of an algorithm in terms of α.

Lemma

Under the GSA, a lattice reduction algorithm has Hermite factor $\zeta = (\alpha^{-1/2})^{n-1}$. □

A simple computation allows us to deduce the Hermite factor of an algorithm in terms of α.

Lemma

Under the GSA, a lattice reduction algorithm has Hermite factor $\zeta = (\alpha^{-1/2})^{n-1}$. □

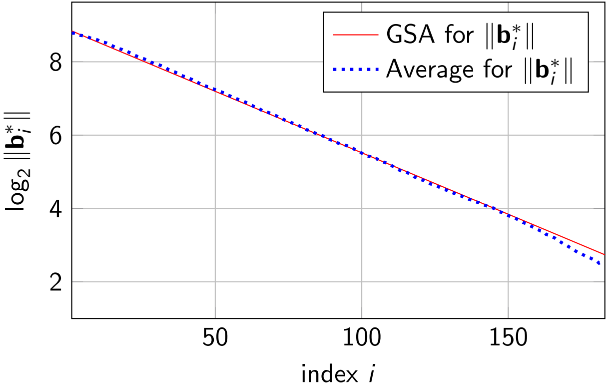


Figure 1: Comparison of a GSA prediction for the profile of a BKZ-56-reduced basis.

Solving SIS via lattice reduction

Solving SIS via lattice reduction

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and let $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ be an SIS instance.

Solving SIS via lattice reduction

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and let $\mathbf{Ax} = \mathbf{0} \bmod q$ be an SIS instance.
- First construct a basis for the right-kernel of $\mathbf{A}$ over $\mathbb{Z}_q$.

Solving SIS via lattice reduction

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and let $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ be an SIS instance.
- First construct a basis for the right-kernel of $\mathbf{A}$ over $\mathbb{Z}_q$.
  - Put $\mathbf{A}$ into reduced echelon form $\begin{bmatrix} \mathbf{I}_n & \hat{\mathbf{A}} \end{bmatrix}$ where $\hat{\mathbf{A}} \in \mathbb{Z}_q^{n \times (m-n)}$.
  - We the obtain a basis of the right-kernel as $\begin{bmatrix} -\hat{\mathbf{A}} \\ \mathbf{I}_{m-n} \end{bmatrix}$ over $\mathbb{Z}_q$.

Solving SIS via lattice reduction

- Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and let $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ be an SIS instance.
- First construct a basis for the right-kernel of $\mathbf{A}$ over $\mathbb{Z}_q$.
  - Put $\mathbf{A}$ into reduced echelon form $\begin{bmatrix} \mathbf{I}_n & \hat{\mathbf{A}} \end{bmatrix}$ where $\hat{\mathbf{A}} \in \mathbb{Z}_q^{n \times (m-n)}$.
  - We the obtain a basis of the right-kernel as $\begin{bmatrix} -\hat{\mathbf{A}} \\ \mathbf{I}_{m-n} \end{bmatrix}$ over $\mathbb{Z}_q$.
- "Lift" this into a row-basis over $\mathbb{Z}$ by defining

$$\mathbf{B} := \begin{bmatrix} -\hat{\mathbf{A}}^T & \mathbf{I}_{m-n} \\ q \cdot \mathbf{I}_n & \mathbf{0} \end{bmatrix} \quad \text{such that} \quad (\mathbf{x}^T, \mathbf{w}^T)\mathbf{B} = (-\mathbf{x}^T\hat{\mathbf{A}}^T + q\mathbf{w}^T, \mathbf{x}^T).$$

Solving SIS via lattice reduction

- Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and let $\mathbf{Ax} = \mathbf{0} \bmod q$ be an SIS instance.
- First construct a basis for the right-kernel of $\mathbf{A}$ over $\mathbb{Z}_q$.
  - Put $\mathbf{A}$ into reduced echelon form $\begin{bmatrix} \mathbf{I}_n & \hat{\mathbf{A}} \end{bmatrix}$ where $\hat{\mathbf{A}} \in \mathbb{Z}_q^{n\times(m-n)}$.
  - We the obtain a basis of the right-kernel as $\begin{bmatrix} -\hat{\mathbf{A}} \\ \mathbf{I}_{m-n} \end{bmatrix}$ over $\mathbb{Z}_q$.
- "Lift" this into a row-basis over $\mathbb{Z}$ by defining

$$\mathbf{B} := \begin{bmatrix} -\hat{\mathbf{A}}^T & \mathbf{I}_{m-n} \\ q\cdot\mathbf{I}_n & \mathbf{0} \end{bmatrix} \quad \text{such that} \quad (\mathbf{x}^T, \mathbf{w}^T)\mathbf{B} = (-\mathbf{x}^T\hat{\mathbf{A}}^T + q\mathbf{w}^T, \mathbf{x}^T).$$

  - We can then verify that

$$\begin{bmatrix} \mathbf{I}_n & \hat{\mathbf{A}} \end{bmatrix} \begin{pmatrix} -\hat{\mathbf{A}}\mathbf{x} + q\mathbf{w} \\ \mathbf{x} \end{pmatrix} = -\hat{\mathbf{A}}\mathbf{x} + q\mathbf{w} + \hat{\mathbf{A}}\mathbf{x} = q\mathbf{w} = \mathbf{0} \bmod q.$$

Solving SIS via lattice reduction

- Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and let $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ be an SIS instance.
- First construct a basis for the right-kernel of $\mathbf{A}$ over $\mathbb{Z}_q$.
  - Put $\mathbf{A}$ into reduced echelon form $\begin{bmatrix} \mathbf{I}_n & \hat{\mathbf{A}} \end{bmatrix}$ where $\hat{\mathbf{A}} \in \mathbb{Z}_q^{n\times(m-n)}$.
  - We the obtain a basis of the right-kernel as $\begin{bmatrix} -\hat{\mathbf{A}} \\ \mathbf{I}_{m-n} \end{bmatrix}$ over $\mathbb{Z}_q$.
- "Lift" this into a row-basis over $\mathbb{Z}$ by defining

$$\mathbf{B} := \begin{bmatrix} -\hat{\mathbf{A}}^T & \mathbf{I}_{m-n} \\ q\cdot\mathbf{I}_n & \mathbf{0} \end{bmatrix} \quad \text{such that} \quad (\mathbf{x}^T, \mathbf{w}^T)\mathbf{B} = (-\mathbf{x}^T\hat{\mathbf{A}}^T + q\mathbf{w}^T, \mathbf{x}^T).$$

  - We can then verify that

$$\begin{bmatrix} \mathbf{I}_n & \hat{\mathbf{A}} \end{bmatrix} \begin{pmatrix} -\hat{\mathbf{A}}\mathbf{x} + q\mathbf{w} \\ \mathbf{x} \end{pmatrix} = -\hat{\mathbf{A}}\mathbf{x} + q\mathbf{w} + \hat{\mathbf{A}}\mathbf{x} = q\mathbf{w} = \mathbf{0} \bmod q.$$

- With a basis $\mathbf{B}$ for our integer lattice, we can then use strong lattice reduction to recover a short vector in the kernel of $\mathbf{A}$, solving $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$.

Solving LWE: the primal attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ be a collection of m LWE samples:
  - $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q = \mathbf{A}\mathbf{s} + \mathbf{e} + q \cdot \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^m$.

Solving LWE: the primal attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ be a collection of m LWE samples:
  - $\mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q = \mathbf{As} + \mathbf{e} + q \cdot \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^m$.
- The primal attack attempts to solve Search-LWE.

Solving LWE: the primal attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ be a collection of m LWE samples:
  - $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q = \mathbf{A}\mathbf{s} + \mathbf{e} + q \cdot \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^m$.
- The primal attack attempts to solve Search-LWE.
- Let $\Lambda_q(\mathbf{A})$ be the *primal lattice* of $\mathbf{A}$:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\ \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q\} \subset \mathbb{Z}^m$$

Solving LWE: the primal attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be a collection of m LWE samples:
  - $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q = \mathbf{A}\mathbf{s} + \mathbf{e} + q \cdot \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^m$.
- The primal attack attempts to solve Search-LWE.
- Let $\Lambda_q(\mathbf{A})$ be the *primal lattice* of $\mathbf{A}$:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\ \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q\} \subset \mathbb{Z}^m$$

- $\mathbf{b}$ is a vector *close* to $\Lambda_q(\mathbf{A})$: $\|\mathbf{b} - (\mathbf{A}\mathbf{s} + q \cdot \mathbf{w})\| = \|\mathbf{e}\|$ where $\mathbf{y} := \mathbf{A}\mathbf{s} + q \cdot \mathbf{w} \in \Lambda_q(\mathbf{A})$.

Solving LWE: the primal attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be a collection of m LWE samples:
  - $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q = \mathbf{A}\mathbf{s} + \mathbf{e} + q \cdot \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^m$.
- The primal attack attempts to solve Search-LWE.
- Let $\Lambda_q(\mathbf{A})$ be the *primal lattice* of $\mathbf{A}$:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\ \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q\} \subset \mathbb{Z}^m$$

- $\mathbf{b}$ is a vector *close* to $\Lambda_q(\mathbf{A})$: $\|\mathbf{b} - (\mathbf{A}\mathbf{s} + q \cdot \mathbf{w})\| = \|\mathbf{e}\|$ where $\mathbf{y} := \mathbf{A}\mathbf{s} + q \cdot \mathbf{w} \in \Lambda_q(\mathbf{A})$.
- By construction, usually $\|\mathbf{e}\| \ll \lambda_1(\Lambda_q(\mathbf{A}))$, meaning this is an instance of BDD.

Solving LWE: the primal attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be a collection of m LWE samples:
  - $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q = \mathbf{A}\mathbf{s} + \mathbf{e} + q \cdot \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^m$.
- The primal attack attempts to solve Search-LWE.
- Let $\Lambda_q(\mathbf{A})$ be the *primal lattice* of $\mathbf{A}$:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\, \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q\} \subset \mathbb{Z}^m$$

- $\mathbf{b}$ is a vector *close* to $\Lambda_q(\mathbf{A})$: $\|\mathbf{b} - (\mathbf{A}\mathbf{s} + q \cdot \mathbf{w})\| = \|\mathbf{e}\|$ where $\mathbf{y} := \mathbf{A}\mathbf{s} + q \cdot \mathbf{w} \in \Lambda_q(\mathbf{A})$.
- By construction, usually $\|\mathbf{e}\| \ll \lambda_1(\Lambda_q(\mathbf{A}))$, meaning this is an instance of BDD.
- From this, $\mathbf{e}$ can be computed as $\mathbf{b} - \mathbf{y}$ and consequently $\mathbf{s}$, (assuming $\mathbf{A}$ has rank n).

Solving LWE: the primal attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be a collection of m LWE samples:
  - $\mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q = \mathbf{As} + \mathbf{e} + q \cdot \mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^m$.
- The primal attack attempts to solve Search-LWE.
- Let $\Lambda_q(\mathbf{A})$ be the *primal lattice* of $\mathbf{A}$:

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\, \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{Ax} \bmod q\} \subset \mathbb{Z}^m$$

- $\mathbf{b}$ is a vector *close* to $\Lambda_q(\mathbf{A})$: $\|\mathbf{b} - (\mathbf{As} + q \cdot \mathbf{w})\| = \|\mathbf{e}\|$ where $\mathbf{y} := \mathbf{As} + q \cdot \mathbf{w} \in \Lambda_q(\mathbf{A})$.
- By construction, usually $\|\mathbf{e}\| \ll \lambda_1(\Lambda_q(\mathbf{A}))$, meaning this is an instance of BDD.
- From this, $\mathbf{e}$ can be computed as $\mathbf{b} - \mathbf{y}$ and consequently $\mathbf{s}$, (assuming $\mathbf{A}$ has rank n).
- Essentially, Search-LWE is an average-case form of BDD.

Solving BDD via lattice reduction

- Let $\mathbf{v} = \mathbf{t} + \mathbf{e} \in \mathrm{span}_{\mathbb{R}}(\mathbf{B})$, where $\mathbf{t} = \mathbf{xB} \in \Lambda(\mathbf{B})$, and let $\mathbf{e}$ be short.

Solving BDD via lattice reduction

- Let $\mathbf{v} = \mathbf{t} + \mathbf{e} \in \mathrm{span}_{\mathbb{R}}(\mathbf{B})$, where $\mathbf{t} = \mathbf{x}\mathbf{B} \in \Lambda(\mathbf{B})$, and let $\mathbf{e}$ be short.
- A classic tecnique by Kannan [17] is to extend $\mathbf{B}$

$$\mathbf{B}' := \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{v} & 1 \end{bmatrix} \text{ such that } \mathbf{v} = \mathbf{x}\mathbf{B} + \mathbf{e} \iff (-\mathbf{x}, 1)\mathbf{B}' = (\mathbf{v} - \mathbf{x}\mathbf{B}, 1) = (\mathbf{e}, 1)$$

where $(\mathbf{e}, 1)$ is still short!

Solving BDD via lattice reduction

- Let $\mathbf{v} = \mathbf{t} + \mathbf{e} \in \text{span}_{\mathbb{R}}(\mathbf{B})$, where $\mathbf{t} = \mathbf{x}\mathbf{B} \in \Lambda(\mathbf{B})$, and let $\mathbf{e}$ be short.
- A classic tecnique by Kannan [17] is to extend $\mathbf{B}$

$$\mathbf{B}' := \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{v} & 1 \end{bmatrix} \text{ such that } \mathbf{v} = \mathbf{x}\mathbf{B} + \mathbf{e} \iff (-\mathbf{x}, 1)\mathbf{B}' = (\mathbf{v} - \mathbf{x}\mathbf{B}, 1) = (\mathbf{e}, 1)$$

  where $(\mathbf{e}, 1)$ is still short!
- And since $\mathbf{e}$ is short enough (BDD), this is an instance of unique-SVP!

Solving BDD via lattice reduction

- Let $\mathbf{v} = \mathbf{t} + \mathbf{e} \in \text{span}_{\mathbb{R}}(\mathbf{B})$, where $\mathbf{t} = \mathbf{x}\mathbf{B} \in \Lambda(\mathbf{B})$, and let $\mathbf{e}$ be short.
- A classic tecnique by Kannan [17] is to extend $\mathbf{B}$

$$\mathbf{B}' := \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{v} & 1 \end{bmatrix} \text{ such that } \mathbf{v} = \mathbf{x}\mathbf{B} + \mathbf{e} \iff (-\mathbf{x}, 1)\mathbf{B}' = (\mathbf{v} - \mathbf{x}\mathbf{B}, 1) = (\mathbf{e}, 1)$$

  where $(\mathbf{e}, 1)$ is still short!
- And since $\mathbf{e}$ is short enough (BDD), this is an instance of unique-SVP!
- Hence by solving uSVP we recover $\mathbf{e}$, from which we recover $\mathbf{t} = \mathbf{v} - \mathbf{e}$, solving BDD.

- Recall that we want to solve BDD for

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\ \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q\} \subset \mathbb{Z}^m$$

- Recall that we want to solve BDD for

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\ \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q\} \subset \mathbb{Z}^m$$

- A possible basis $\mathbf{B}$ for $\Lambda_q(\mathbf{A})$ is

$$\mathbf{B} := \begin{bmatrix} q \cdot \mathbf{I}_m \\ \mathbf{A}^T \end{bmatrix} \text{ such that } \mathbf{y}^T = (\mathbf{w}^T, \mathbf{x}^T)\mathbf{B} = q \cdot \mathbf{w}^T + \mathbf{x}^T\mathbf{A}^T \iff \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q$$

- Recall that we want to solve BDD for

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists\, \mathbf{x} \in \mathbb{Z}^n \text{ such that } \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q\} \subset \mathbb{Z}^m$$

- A possible basis $\mathbf{B}$ for $\Lambda_q(\mathbf{A})$ is

$$\mathbf{B} := \begin{bmatrix} q \cdot \mathbf{I}_m \\ \mathbf{A}^T \end{bmatrix} \quad \text{such that} \quad \mathbf{y}^T = (\mathbf{w}^T, \mathbf{x}^T)\mathbf{B} = q \cdot \mathbf{w}^T + \mathbf{x}^T\mathbf{A}^T \iff \mathbf{y} = \mathbf{A}\mathbf{x} \bmod q$$

- Using Kannan's embedding,

$$\mathbf{B}' := \begin{bmatrix} q \cdot \mathbf{I}_m & \mathbf{0} \\ \mathbf{A}^T & \mathbf{0} \\ \mathbf{b}^T & 1 \end{bmatrix} \quad \text{such that} \quad (-\mathbf{w}^T, -\mathbf{s}^T, 1)\mathbf{B}' = (\mathbf{b}^T - \mathbf{s}^T\mathbf{A}^T - q \cdot \mathbf{w}^T, 1) = (\mathbf{e}^T, 1).$$

Overview

- To solve Search-LWE, a solution is to perform strong lattice reduction on $\mathbf{B}'$ to recover the unusually short "target" vector $(\mathbf{e}, 1)$.

Overview

- To solve Search-LWE, a solution is to perform strong lattice reduction on $\mathbf{B}'$ to recover the unusually short "target" vector $(\mathbf{e}, 1)$.
- Unlike SIS, $\mathbf{e}$ was "planted".

Overview

- To solve Search-LWE, a solution is to perform strong lattice reduction on $\mathbf{B}'$ to recover the unusually short "target" vector $(\mathbf{e}, 1)$.
- Unlike SIS, $\mathbf{e}$ was "planted".
- Meaning $\Lambda(\mathbf{B}')$ is not quite "random q-ary".

Overview

- To solve Search-LWE, a solution is to perform strong lattice reduction on $\mathbf{B}'$ to recover the unusually short "target" vector $(\mathbf{e}, 1)$.
- Unlike SIS, $\mathbf{e}$ was "planted".
- Meaning $\Lambda(\mathbf{B}')$ is not quite "random q-ary".
    - If lattice reduction is strong enough, projections $\pi_k(\mathbf{e}, 1)$ will not respect the GSA.

Overview

- To solve Search-LWE, a solution is to perform strong lattice reduction on $\mathbf{B}'$ to recover the unusually short "target" vector $(\mathbf{e}, 1)$.
- Unlike SIS, $\mathbf{e}$ was "planted".
- Meaning $\Lambda(\mathbf{B}')$ is not quite "random q-ary".
  - If lattice reduction is strong enough, projections $\pi_k(\mathbf{e}, 1)$ will not respect the GSA.
  - Eventually, $\pi_k(\mathbf{e}, 1)$ can be recovered as $\mathbf{b}_i^*$ and recovery of $(\mathbf{e}, 1)$ becomes easy [18].

Solving LWE: the dual attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$
  - Either $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$ or $\mathbf{b} \sim U(\mathbb{Z}_q^m)$.

Solving LWE: the dual attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$
  - Either $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$ or $\mathbf{b} \sim U(\mathbb{Z}_q^m)$.
- The *dual attack* solves Decision-LWE.

Solving LWE: the dual attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$
  - Either $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$ or $\mathbf{b} \sim U(\mathbb{Z}_q^m)$.
- The *dual attack* solves Decision-LWE.
- Let the *dual lattice* $\Lambda_q^{\perp}(\mathbf{A})$ be

$$\Lambda_q^{\perp}(\mathbf{A}) := \left\{ \mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}^T \mathbf{x} = \mathbf{0} \bmod q \right\} \subset \mathbb{Z}^m$$

Solving LWE: the dual attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$
  - Either $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$ or $\mathbf{b} \sim U(\mathbb{Z}_q^m)$.
- The *dual attack* solves Decision-LWE.
- Let the *dual lattice* $\Lambda_q^\perp(\mathbf{A})$ be

$$\Lambda_q^\perp(\mathbf{A}) := \left\{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}^T\mathbf{x} = \mathbf{0} \bmod q\right\} \subset \mathbb{Z}^m$$

- The dual attack searches for a short $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$, and uses it to distinguish "LWE" or "uniform"

Solving LWE: the dual attack

- Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$
  - Either $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$ or $\mathbf{b} \sim U(\mathbb{Z}_q^m)$.
- The *dual attack* solves Decision-LWE.
- Let the *dual lattice* $\Lambda_q^\perp(\mathbf{A})$ be

$$\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}^T\mathbf{x} = \mathbf{0} \bmod q\} \subset \mathbb{Z}^m$$

- The dual attack searches for a short $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$, and uses it to distinguish "LWE" or "uniform"
- Essentially, this is an average-case instance of SIS
  - To find $\mathbf{x}$, follow the SIS methodology.

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \left\|x^T\mathbf{b}\right\| = \left\|\mathbf{x}^T\mathbf{e}\right\| \leq \|x\|\,\|e\| \ll q$

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \left\|x^T\mathbf{b}\right\| = \left\|\mathbf{x}^T\mathbf{e}\right\| \leq \|x\| \|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \left\|x^T\mathbf{b}\right\| = \left\|\mathbf{x}^T\mathbf{e}\right\| \leq \|x\| \, \|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \left\|x^T\mathbf{b}\right\| = \left\|\mathbf{x}^T\mathbf{e}\right\| \leq \|x\| \, \|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

Accuracy/cost trade-off

- Let $b = 0$ if "LWE", and $b = 1$ if "uniform".

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \left\|x^T\mathbf{b}\right\| = \left\|\mathbf{x}^T\mathbf{e}\right\| \leq \|x\| \, \|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

Accuracy/cost trade-off

- Let $b = 0$ if "LWE", and $b = 1$ if "uniform".
- Re-test N times by re-randomizing the basis of $\Lambda_q^\perp(\mathbf{A})$.

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \left\|x^T\mathbf{b}\right\| = \left\|\mathbf{x}^T\mathbf{e}\right\| \leq \|x\|\,\|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

Accuracy/cost trade-off

- Let $b = 0$ if "LWE", and $b = 1$ if "uniform".
- Re-test N times by re-randomizing the basis of $\Lambda_q^\perp(\mathbf{A})$.
- Let $T_i \in \{0, 1\}$, for $1 \leq i \leq N$, be the outcome of the i-th run.

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \left\|x^T\mathbf{b}\right\| = \left\|\mathbf{x}^T\mathbf{e}\right\| \leq \|x\|\,\|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

Accuracy/cost trade-off

- Let $b = 0$ if "LWE", and $b = 1$ if "uniform".
- Re-test N times by re-randomizing the basis of $\Lambda_q^{\perp}(\mathbf{A})$.
- Let $T_i \in \{0, 1\}$, for $1 \leq i \leq N$, be the outcome of the i-th run.
- Suppose $\Pr[T_i = b] \geq \frac{1}{2} + \varepsilon$.

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \|x^T\mathbf{b}\| = \|\mathbf{x}^T\mathbf{e}\| \leq \|x\|\,\|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

Accuracy/cost trade-off

- Let $b = 0$ if "LWE", and $b = 1$ if "uniform".
- Re-test N times by re-randomizing the basis of $\Lambda_q^\perp(\mathbf{A})$.
- Let $T_i \in \{0, 1\}$, for $1 \leq i \leq N$, be the outcome of the i-th run.
- Suppose $\Pr[T_i = b] \geq \frac{1}{2} + \varepsilon$.
- After N tests, guess $b = \lfloor \sum_{i=1}^{N} T_i/N \rceil$.

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \|x^T\mathbf{b}\| = \|\mathbf{x}^T\mathbf{e}\| \leq \|x\|\,\|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

Accuracy/cost trade-off

- Let $b = 0$ if "LWE", and $b = 1$ if "uniform".
- Re-test N times by re-randomizing the basis of $\Lambda_q^{\perp}(\mathbf{A})$.
- Let $T_i \in \{0, 1\}$, for $1 \leq i \leq N$, be the outcome of the i-th run.
- Suppose $\Pr[T_i = b] \geq \frac{1}{2} + \varepsilon$.
- After N tests, guess $b = \lfloor \sum_{i=1}^{N} T_i/N \rceil$.
- Using the Chernoff bound over $\sum T_i$, setting $N = 1/(2\varepsilon^2)$ bounds $\Pr[wrong] \leq (2/e) \cdot 2^{-N}$.

- Once such $\mathbf{x}$ is found, compute $\mathbf{x}^T\mathbf{b}$:
  - LWE case: $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \cancel{(\mathbf{x}^T\mathbf{A})}\mathbf{s} + \mathbf{x}^T\mathbf{e} \bmod q \implies \|x^T\mathbf{b}\| = \|\mathbf{x}^T\mathbf{e}\| \leq \|x\|\,\|e\| \ll q$
  - Uniform case: $\mathbf{x}^T\mathbf{b}$ will be uniformly distributed
- This is a probabilistic distinguisher: the shorter $\mathbf{x}$ the better the guess.
  - But the harder to find $\mathbf{x}$.

Accuracy/cost trade-off

- Let $b = 0$ if "LWE", and $b = 1$ if "uniform".
- Re-test N times by re-randomizing the basis of $\Lambda_q^{\perp}(\mathbf{A})$.
- Let $T_i \in \{0, 1\}$, for $1 \leq i \leq N$, be the outcome of the i-th run.
- Suppose $\Pr[T_i = b] \geq \frac{1}{2} + \varepsilon$.
- After N tests, guess $b = \lfloor \sum_{i=1}^{N} T_i/N \rceil$.
- Using the Chernoff bound over $\sum T_i$, setting $N = 1/(2\varepsilon^2)$ bounds $\Pr[wrong] \leq (2/e) \cdot 2^{-N}$.
- Harder reduction $\iff$ smaller $\mathbf{x}$ $\iff$ larger ε $\iff$ smaller N!

Solving LWE: non-lattice approaches

- It should be noted that all these attacks use classical algorithms.

Solving LWE: non-lattice approaches

- It should be noted that all these attacks use classical algorithms.
- Quantum attacks can be derived using Grover's algorithm as a black-box [19].
  - Likely these would not *actually* work, even with a QC!

Solving LWE: non-lattice approaches

- It should be noted that all these attacks use classical algorithms.
- Quantum attacks can be derived using Grover's algorithm as a black-box [19].
  - Likely these would not *actually* work, even with a QC!
- There exist a few non-lattice approaches to solving LWE.

Solving LWE: non-lattice approaches

- It should be noted that all these attacks use classical algorithms.
- Quantum attacks can be derived using Grover's algorithm as a black-box [19].
  - Likely these would not *actually* work, even with a QC!
- There exist a few non-lattice approaches to solving LWE.

Algebraic attacks: Arora-Ge [20]

- Set up a polynomial system over $\mathbb{Z}_q$,

$$\left\{ \prod_{\eta \in \mathsf{Supp}(\chi_e)} (\mathbf{b}_i - \langle \mathbf{a}_i, \mathbf{s} \rangle - \eta) = 0 \right\}_i$$

Solving LWE: non-lattice approaches

- It should be noted that all these attacks use classical algorithms.
- Quantum attacks can be derived using Grover's algorithm as a black-box [19].
  - Likely these would not *actually* work, even with a QC!
- There exist a few non-lattice approaches to solving LWE.

Algebraic attacks: Arora-Ge [20]

- Set up a polynomial system over $\mathbb{Z}_q$,

$$\left\{ \prod_{\eta \in \mathsf{Supp}(\chi_e)} (\mathbf{b}_i - \langle \mathbf{a}_i, \mathbf{s} \rangle - \eta) = 0 \right\}_i$$

- Use F4/Groebner bases to solve for $\mathbf{s}$.

Solving LWE: non-lattice approaches

- It should be noted that all these attacks use classical algorithms.
- Quantum attacks can be derived using Grover's algorithm as a black-box [19].
  - Likely these would not *actually* work, even with a QC!
- There exist a few non-lattice approaches to solving LWE.

Algebraic attacks: Arora-Ge [20]

- Set up a polynomial system over $\mathbb{Z}_q$,

$$\left\{ \prod_{\eta \in \mathrm{Supp}(\chi_e)} (\mathbf{b}_i - \langle \mathbf{a}_i, \mathbf{s} \rangle - \eta) = 0 \right\}_i$$

- Use F4/Groebner bases to solve for **s**.
- Asymptotically the best approach whenever χ_e has width $O(\sqrt{n})$!
- Practically inefficient

Solving LWE: non-lattice approaches

Combinatorial attacks: BKW [21].

- Borrowed from the Learning Parity with Noise (LPN) literature.

Solving LWE: non-lattice approaches

Combinatorial attacks: BKW [21].

- Borrowed from the Learning Parity with Noise (LPN) literature.
- Solves LWE by combining LWE samples until they produce some leading to easy secret recovery.

Solving LWE: non-lattice approaches

Combinatorial attacks: BKW [21].

- Borrowed from the Learning Parity with Noise (LPN) literature.
- Solves LWE by combining LWE samples until they produce some leading to easy secret recovery.
- BKW variants [22], [23] result in a subexponential-time algorithm against LWE with binary secret, $\chi_s = U(\{0,1\})$.

Solving LWE: non-lattice approaches

Combinatorial attacks: BKW [21].

- Borrowed from the Learning Parity with Noise (LPN) literature.
- Solves LWE by combining LWE samples until they produce some leading to easy secret recovery.
- BKW variants [22], [23] result in a subexponential-time algorithm against LWE with binary secret, $\chi_s = U(\{0, 1\})$.
- However, BKW requires access to a number of samples m larger than usually available in the cryptographic setting.

Solving LWE: non-lattice approaches

Combinatorial attacks: BKW [21].

- Borrowed from the Learning Parity with Noise (LPN) literature.
- Solves LWE by combining LWE samples until they produce some leading to easy secret recovery.
- BKW variants [22], [23] result in a subexponential-time algorithm against LWE with binary secret, $\chi_s = U(\{0, 1\})$.
- However, BKW requires access to a number of samples m larger than usually available in the cryptographic setting.
- Overall, an impractical attack.

In conclusion

In conclusion

- We've introduced basic definitions and results on real lattices.

In conclusion

- We've introduced basic definitions and results on real lattices.
- We've seen how lattice problems can be turned into useful hard-on-average hardness assumptions.

In conclusion

- We've introduced basic definitions and results on real lattices.
- We've seen how lattice problems can be turned into useful hard-on-average hardness assumptions.
- We've also seen that the connection is not just theoretical, thanks to lattice reduction.

In conclusion

- We've introduced basic definitions and results on real lattices.
- We've seen how lattice problems can be turned into useful hard-on-average hardness assumptions.
- We've also seen that the connection is not just theoretical, thanks to lattice reduction.

In conclusion

- We've introduced basic definitions and results on real lattices.
- We've seen how lattice problems can be turned into useful hard-on-average hardness assumptions.
- We've also seen that the connection is not just theoretical, thanks to lattice reduction.

Thank you

Resources I

[1] V. Vaikuntanathan, *Lattices, learning with errors and post-quantum cryptography*, https://people.csail.mit.edu/vinodv/CS294/, Accessed: 2025-07-25, 2020.

[2] F. Virdia, "Post-quantum cryptography: Cryptanalysis and implementation," English, Ph.D. dissertation, Royal Holloway, University of London, 2021.

[3] C. Hermite, "Extraits de lettres de m. ch. hermite à m. jacobi sur différents objects de la théorie des nombres. (continuation).," fre, *Journal für die reine und angewandte Mathematik*, vol. 40, pp. 279–315, 1850. [Online]. Available: http://eudml.org/doc/147463.

[4] J. Martinet, *Perfect Lattices in Euclidean Spaces*. Springer Berlin Heidelberg, 2003. DOI: 10.1007/978-3-662-05167-2. [Online]. Available: https://doi.org/10.1007/978-3-662-05167-2.

Resources II

[5] L. J. Mordell, "Observation on the minimum of a positive quadratic form in eight variables," *Journal of the London Mathematical Society*, vol. 19, no. 73_Part_1, pp. 3–6, 1944.

[6] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. Wiley, Oct. 1968, ISBN: 0471257087. [Online]. Available: https://www.xarg.org/ref/a/0471257087/.

[7] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-Quantum Cryptography*, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds., Berlin, Heidelberg, New York: Springer, Heidelberg, 2009, pp. 147–191.

[8] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *CRYPTO 2009*, S. Halevi, Ed., ser. LNCS, vol. 5677, Springer, Berlin, Heidelberg, Aug. 2009, pp. 595–618. DOI: 10.1007/978-3-642-03356-8_35.

Resources III

[9] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on gaussian measures," *SIAM journal on computing*, vol. 37, no. 1, pp. 267–302, 2007.

[10] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, Sep. 2009, ISSN: 0004-5411. DOI: 10.1145/1568318.1568324. [Online]. Available: https://doi.org/10.1145/1568318.1568324.

[11] S. Chatterjee, N. Koblitz, A. Menezes, and P. Sarkar, *Another look at tightness II: Practical issues in cryptography*, Cryptology ePrint Archive, Paper 2016/360, 2016. [Online]. Available: https://eprint.iacr.org/2016/360.

[12] P. Q. Nguyen and B. Vallée, Eds., *The LLL Algorithm - Survey and Applications* (Information Security and Cryptography). Springer, 2010, ISBN: 978-3-642-02294-4. DOI: 10.1007/978-3-642-02295-1. [Online]. Available: https://doi.org/10.1007/978-3-642-02295-1.

Resources IV

[13] P. Q. Nguyen and D. Stehlé, “Lll on the average,” in *Algorithmic Number Theory*, F. Hess, S. Pauli, and M. Pohst, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 238–256, ISBN: 978-3-540-36076-6.

[14] N. Gama and P. Q. Nguyen, “Predicting lattice reduction,” in *EUROCRYPT 2008*, N. P. Smart, Ed., ser. LNCS, vol. 4965, Springer, Berlin, Heidelberg, Apr. 2008, pp. 31–51. DOI: 10.1007/978-3-540-78967-3_3.

[15] Y. Chen, “Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe,” Available at https://archive.org/details/PhDChen13, Ph.D. dissertation, Paris 7, 2013. [Online]. Available: https://archive.org/details/PhDChen13.

Resources V

[16] C. Schnorr, “Lattice reduction by random sampling and birthday methods,” in *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, H. Alt and M. Habib, Eds., ser. Lecture Notes in Computer Science, vol. 2607, Springer, 2003, pp. 145–156. DOI: 10.1007/3-540-36494-3_14. [Online]. Available: http://dx.doi.org/10.1007/3-540-36494-3_14.

[17] R. Kannan, “Minkowski's convex body theorem and integer programming,” *Mathematics of Operations Research*, vol. 12, no. 3, pp. 415–440, Aug. 1987, ISSN: 1526-5471. DOI: 10.1287/moor.12.3.415.

[18] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, “Revisiting the expected cost of solving uSVP and applications to LWE,” in *ASIACRYPT 2017, Part I*, T. Takagi and T. Peyrin, Eds., ser. LNCS, vol. 10624, Springer, Cham, Dec. 2017, pp. 297–322. DOI: 10.1007/978-3-319-70694-8_11.

Resources VI

[19] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *28th ACM STOC*, ACM Press, May 1996, pp. 212–219. DOI: 10.1145/237814.237866.

[20] S. Arora and R. Ge, "New algorithms for learning in presence of errors," in *ICALP 2011, Part I*, L. Aceto, M. Henzinger, and J. Sgall, Eds., ser. LNCS, vol. 6755, Springer, Berlin, Heidelberg, Jul. 2011, pp. 403–415. DOI: 10.1007/978-3-642-22006-7_34.

[21] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," in *32nd ACM STOC*, ACM Press, May 2000, pp. 435–440. DOI: 10.1145/335305.335355.

[22] P. Kirchner and P.-A. Fouque, "An improved BKW algorithm for LWE with applications to cryptography and lattices," in *CRYPTO 2015, Part I*, R. Gennaro and M. J. B. Robshaw, Eds., ser. LNCS, vol. 9215, Springer, Berlin, Heidelberg, Aug. 2015, pp. 43–62. DOI: 10.1007/978-3-662-47989-6_3.

Resources VII

[23] Q. Guo, T. Johansson, and P. Stankovski, "Coded-BKW: Solving LWE using lattice codes," in *CRYPTO 2015, Part I*, R. Gennaro and M. J. B. Robshaw, Eds., ser. LNCS, vol. 9215, Springer, Berlin, Heidelberg, Aug. 2015, pp. 23–42. DOI: 10.1007/978-3-662-47989-6_2.