The Geometry of Lattice Cryptography

Daniele Micciancio

Department of Computer Science and Engineering
University of California, San Diego

August 29-30, 2011 (FOSAD '11 – Bertinoro, Italy)

Cryptography, Complexity and Lattices

Cryptography: exploiting hard computational problems to build computer systems that are hard to break.

Good news

There are plenty of hard computational problems in computer science.

Bad news

Finding cryptographically useful hard problems seems hard.

Cryptography requires problems that

- are very hard to solve: solution should take enormous time
- are hard to solve on average, even with small probability
- have extra features, e.g., trapdoors, regularity, etc.

Cryptography, Complexity and Lattices

Cryptography: exploiting hard computational problems to build computer systems that are hard to break.

Good news

There are plenty of hard computational problems in computer science.

Bad news

Finding cryptographically useful hard problems seems hard.

Cryptography requires problems that

- are very hard to solve: solution should take enormous time
- are hard to solve on average, even with small probability
- have extra features, e.g., trapdoors, regularity, etc.

Cryptography, Complexity and Lattices

Cryptography: exploiting hard computational problems to build computer systems that are hard to break.

Good news

There are plenty of hard computational problems in computer science.

Bad news

Finding cryptographically useful hard problems seems hard.

Cryptography requires problems that

- are very hard to solve: solution should take enormous time
- are hard to solve on average, even with small probability
- have extra features, e.g., trapdoors, regularity, etc.

Cryptography, Complexity and Lattices

Cryptography: exploiting hard computational problems to build computer systems that are hard to break.

Good news

There are plenty of hard computational problems in computer science.

Bad news

Finding cryptographically useful hard problems seems hard.

Cryptography requires problems that

- are very hard to solve: solution should take enormous time
- are hard to solve on average, even with small probability
- have extra features, e.g., trapdoors, regularity, etc.

Cryptography, Complexity and Lattices

Cryptography: exploiting hard computational problems to build computer systems that are hard to break.

Good news

There are plenty of hard computational problems in computer science.

Bad news

Finding cryptographically useful hard problems seems hard.

Cryptography requires problems that

- are very hard to solve: solution should take enormous time
- are hard to solve on average, even with small probability
- have extra features, e.g., trapdoors, regularity, etc.

Cryptography, Complexity and Lattices

Cryptography: exploiting hard computational problems to build computer systems that are hard to break.

Good news

There are plenty of hard computational problems in computer science.

Bad news

Finding cryptographically useful hard problems seems hard.

Cryptography requires problems that

- are very hard to solve: solution should take enormous time
- are hard to solve on average, even with small probability
- have extra features, e.g., trapdoors, regularity, etc.

Point Lattices and Cryptography

Lattice problems

- appear to be very hard (solution takes exponential time),
- have been widely studied by mathematicians since 19th century (Lagrange, Gauss, Dirichlet, . . .),
- provably yield hard on average problems, from worst-case complexity assumptions.

Lattice related constructions and cryptographic functions

- have many useful features (linearity, trapdoors, etc.),
- are efficient and easy to implement, typically involving only simple arithmetic operations on small numbers.

Point Lattices and Cryptography

Lattice problems

- appear to be very hard (solution takes exponential time),
- have been widely studied by mathematicians since 19th century (Lagrange, Gauss, Dirichlet, . . .),
- provably yield hard on average problems, from worst-case complexity assumptions.

Lattice related constructions and cryptographic functions

- have many useful features (linearity, trapdoors, etc.),
- are efficient and easy to implement, typically involving only simple arithmetic operations on small numbers.

Point Lattices and Cryptography

Lattice problems

- appear to be very hard (solution takes exponential time),
- have been widely studied by mathematicians since 19th century (Lagrange, Gauss, Dirichlet, . . .),
- provably yield hard on average problems, from worst-case complexity assumptions.

Lattice related constructions and cryptographic functions

- have many useful features (linearity, trapdoors, etc.),
- are efficient and easy to implement, typically involving only simple arithmetic operations on small numbers.

Point Lattices and Cryptography

Lattice problems

- appear to be very hard (solution takes exponential time),
- have been widely studied by mathematicians since 19th century (Lagrange, Gauss, Dirichlet, . . .),
- provably yield hard on average problems, from worst-case complexity assumptions.

Lattice related constructions and cryptographic functions

- have many useful features (linearity, trapdoors, etc.),
- are efficient and easy to implement, typically involving only simple arithmetic operations on small numbers.

Point Lattices and Cryptography

Lattice problems

- appear to be very hard (solution takes exponential time),
- have been widely studied by mathematicians since 19th century (Lagrange, Gauss, Dirichlet, . . .),
- provably yield hard on average problems, from worst-case complexity assumptions.

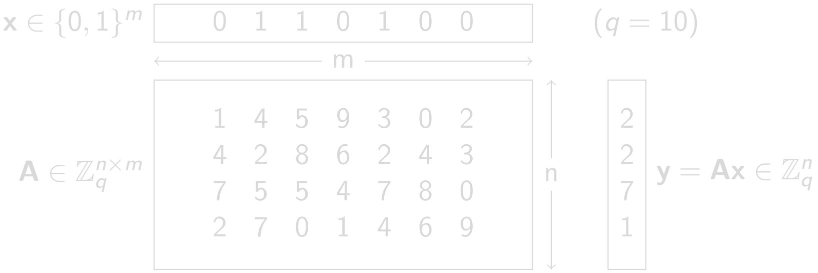
Lattice related constructions and cryptographic functions

- have many useful features (linearity, trapdoors, etc.),
- are efficient and easy to implement, typically involving only simple arithmetic operations on small numbers.

Ajtai's function

Definition (Ajtai's function)

$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q \qquad$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{x} \in \{0,1\}^m$



$\mathbf{x} \in \{0,1\}^m$ $\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$ $(q = 10)$

$\xleftrightarrow{\quad m \quad}$

$$\mathbf{A} \in \mathbb{Z}_q^{n\times m} \quad \begin{bmatrix} 1 & 4 & 5 & 9 & 3 & 0 & 2 \\ 4 & 2 & 8 & 6 & 2 & 4 & 3 \\ 7 & 5 & 5 & 4 & 7 & 8 & 0 \\ 2 & 7 & 0 & 1 & 4 & 6 & 9 \end{bmatrix} \; n \quad \begin{bmatrix} 2 \\ 2 \\ 7 \\ 1 \end{bmatrix} \; \mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$$

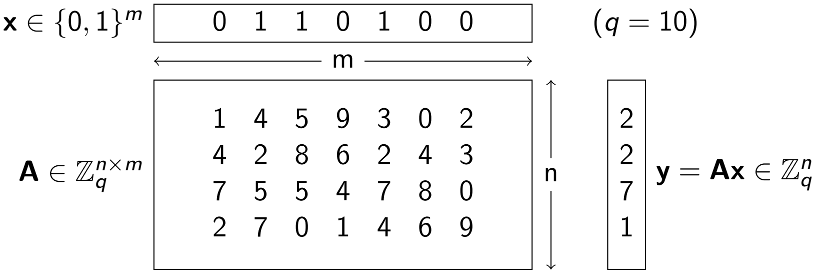
Security (One-wayness)

Given $\mathbf{A}$ and $\mathbf{y}$, it is hard to find $\mathbf{x} \in \{0,1\}^m$ s.t. $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{y}$.

Ajtai's function

Definition (Ajtai's function)

$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q \qquad \text{where } \mathbf{A} \in \mathbb{Z}_q^{n\times m} \text{ and } \mathbf{x} \in \{0,1\}^m$



$\mathbf{x} \in \{0,1\}^m$ $\quad \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$ $\quad (q = 10)$

$\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ $\quad \begin{bmatrix} 1 & 4 & 5 & 9 & 3 & 0 & 2 \\ 4 & 2 & 8 & 6 & 2 & 4 & 3 \\ 7 & 5 & 5 & 4 & 7 & 8 & 0 \\ 2 & 7 & 0 & 1 & 4 & 6 & 9 \end{bmatrix}$ $\quad \begin{bmatrix} 2 \\ 2 \\ 7 \\ 1 \end{bmatrix}$ $\quad \mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$

Security (One-wayness)

Given $\mathbf{A}$ and $\mathbf{y}$, it is hard to find $\mathbf{x} \in \{0,1\}^m$ s.t. $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{y}$.

Ajtai's function

Definition (Ajtai's function)

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q \qquad \text{where } \mathbf{A} \in \mathbb{Z}_q^{n\times m} \text{ and } \mathbf{x} \in \{0,1\}^m$$

$$\mathbf{x} \in \{0,1\}^m \quad \begin{array}{|ccccccc|}\hline 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hline \end{array} \qquad (q = 10)$$

$$\xleftrightarrow{\quad\quad\quad\quad\text{m}\quad\quad\quad\quad}$$

$$\mathbf{A} \in \mathbb{Z}_q^{n\times m} \quad \begin{array}{|ccccccc|}\hline 1 & 4 & 5 & 9 & 3 & 0 & 2 \\ 4 & 2 & 8 & 6 & 2 & 4 & 3 \\ 7 & 5 & 5 & 4 & 7 & 8 & 0 \\ 2 & 7 & 0 & 1 & 4 & 6 & 9 \\ \hline \end{array} \ \updownarrow \text{n} \qquad \begin{array}{|c|}\hline 2 \\ 2 \\ 7 \\ 1 \\ \hline \end{array} \ \mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$$

Security (One-wayness)

Given $\mathbf{A}$ and $\mathbf{y}$, it is hard to find $\mathbf{x} \in \{0,1\}^m$ s.t. $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{y}$.

Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

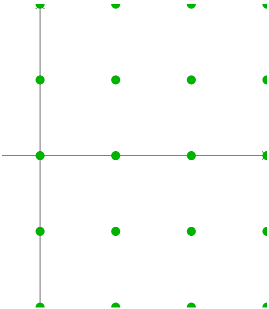
Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

Point Lattices

- The simplest example of lattice is $\mathbb{Z}^n = \{(x_1, \ldots, x_n) \colon x_i \in \mathbb{Z}\}$
- Other lattices are obtained by applying a linear transformation

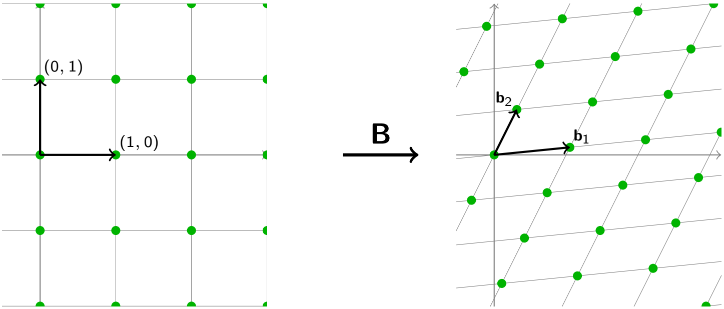
$$\mathbf{B} \colon \mathbf{x} = (x_1, \ldots, x_n) \mapsto \mathbf{Bx} = x_1 \cdot \mathbf{b}_1 + \cdots + x_n \cdot \mathbf{b}_n$$



Point Lattices

- The simplest example of lattice is $\mathbb{Z}^n = \{(x_1, \ldots, x_n) \colon x_i \in \mathbb{Z}\}$
- Other lattices are obtained by applying a linear transformation

$$\mathbf{B} \colon \mathbf{x} = (x_1, \ldots, x_n) \mapsto \mathbf{B}\mathbf{x} = x_1 \cdot \mathbf{b}_1 + \cdots + x_n \cdot \mathbf{b}_n$$



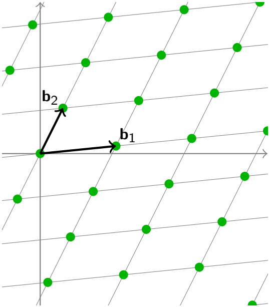
Lattices and Bases

A lattice is the set of all integer linear combinations of (linearly independent) basis vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$:

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{b}_i \cdot \mathbb{Z} = \{\mathbf{B}\mathbf{x} \colon \mathbf{x} \in \mathbb{Z}^n\}$$

The same lattice has many bases

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{c}_i \cdot \mathbb{Z}$$



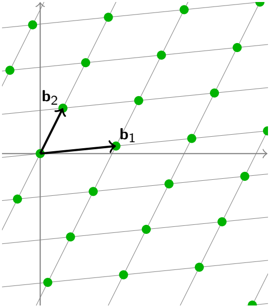
Lattices and Bases

A lattice is the set of all integer linear combinations of (linearly independent) basis vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$:

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{b}_i \cdot \mathbb{Z} = \{\mathbf{B}\mathbf{x} \colon \mathbf{x} \in \mathbb{Z}^n\}$$

The same lattice has many bases

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{c}_i \cdot \mathbb{Z}$$



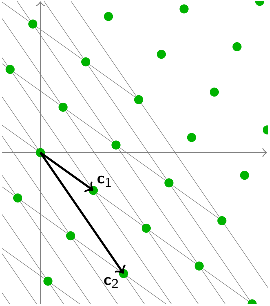
Lattices and Bases

A lattice is the set of all integer linear combinations of (linearly independent) basis vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$:

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{b}_i \cdot \mathbb{Z} = \{\mathbf{B}\mathbf{x} \colon \mathbf{x} \in \mathbb{Z}^n\}$$

The same lattice has many bases

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{c}_i \cdot \mathbb{Z}$$



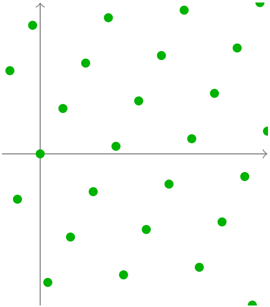
Lattices and Bases

A lattice is the set of all integer linear combinations of (linearly independent) basis vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$:

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{b}_i \cdot \mathbb{Z} = \{\mathbf{Bx} \colon \mathbf{x} \in \mathbb{Z}^n\}$$

The same lattice has many bases

$$\mathcal{L} = \sum_{i=1}^{n} \mathbf{c}_i \cdot \mathbb{Z}$$



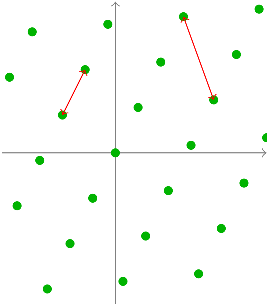
Definition (Lattice)

A discrete additive subgroup of $\mathbb{R}^n$

Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| = \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$



- Successive minima ($i = 1, \ldots, n$)

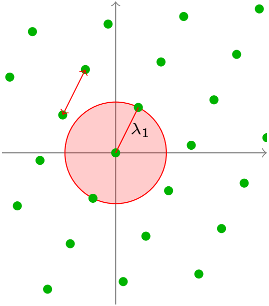
$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r) \cap \mathcal{L}) \geq i\}$$

- Examples
  - $\mathbb{Z}^n$: $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$
  - Always: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\|$$
$$= \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$



- Successive minima ($i = 1,\ldots,n$)

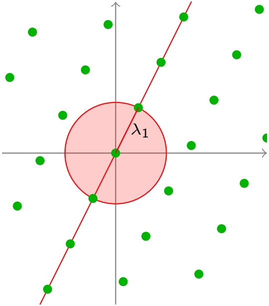
$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r)\cap\mathcal{L}) \geq i\}$$

- Examples
  - $\mathbb{Z}^n$: $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$
  - Always: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

Minimum Distance and Successive Minima

- Minimum distance

$$\begin{aligned}\lambda_1 &= \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| \\ &= \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|\end{aligned}$$



- Successive minima ($i = 1, \ldots, n$)

$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r) \cap \mathcal{L}) \geq i\}$$

- Examples
  - $\mathbb{Z}^n$: $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$
  - Always: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

Minimum Distance and Successive Minima

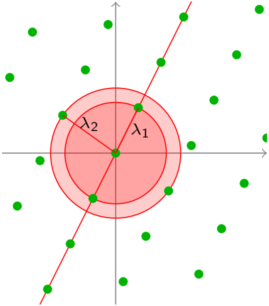
- Minimum distance

$$\begin{aligned}\lambda_1 &= \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| \\ &= \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|\end{aligned}$$

- Successive minima ($i = 1, \ldots, n$)

$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r)\cap\mathcal{L}) \geq i\}$$

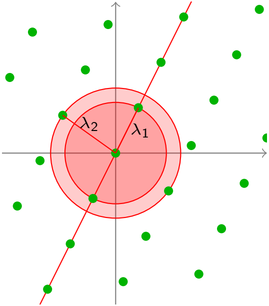
- Examples
  - $\mathbb{Z}^n$: $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$
  - Always: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$



Minimum Distance and Successive Minima

- Minimum distance

$$\begin{aligned}\lambda_1 &= \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| \\ &= \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|\end{aligned}$$



- Successive minima ($i = 1, \ldots, n$)

$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r)\cap\mathcal{L}) \geq i\}$$

- Examples
  - $\mathbb{Z}^n$: $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$
  - Always: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

Distance Function and Covering Radius

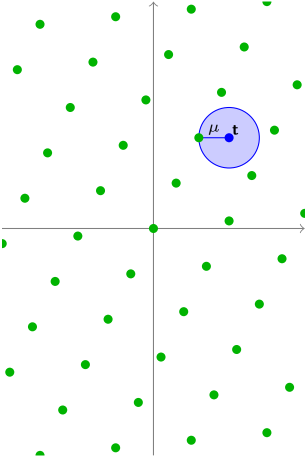
- Distance function

$$\mu(\mathbf{t}, \mathcal{L}) = \min_{\mathbf{x} \in \mathcal{L}} \|\mathbf{t} - \mathbf{x}\|$$

- Covering radius

$$\mu(\mathcal{L}) = \max_{\mathbf{t} \in span(\mathcal{L})} \mu(\mathbf{t}, \mathcal{L})$$

- Spheres or radius $\mu(\mathcal{L})$ centered around all lattice points cover the whole space



Distance Function and Covering Radius

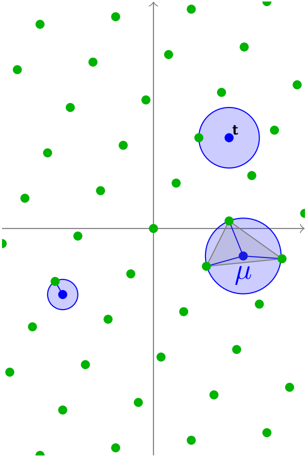
- Distance function

$$\mu(\mathbf{t},\mathcal{L}) = \min_{\mathbf{x}\in\mathcal{L}} \|\mathbf{t}-\mathbf{x}\|$$

- Covering radius

$$\mu(\mathcal{L}) = \max_{\mathbf{t}\in span(\mathcal{L})} \mu(\mathbf{t},\mathcal{L})$$

- Spheres or radius $\mu(\mathcal{L})$ centered around all lattice points cover the whole space



Distance Function and Covering Radius

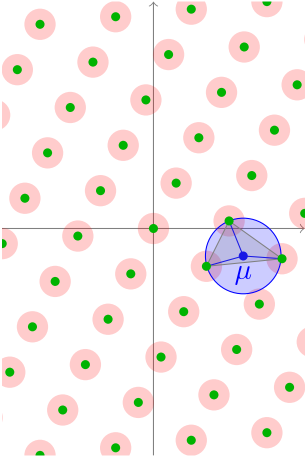
- Distance function

$$\mu(\mathbf{t},\mathcal{L}) = \min_{\mathbf{x}\in\mathcal{L}} \|\mathbf{t}-\mathbf{x}\|$$

- Covering radius

$$\mu(\mathcal{L}) = \max_{\mathbf{t}\in span(\mathcal{L})} \mu(\mathbf{t},\mathcal{L})$$

- Spheres or radius $\mu(\mathcal{L})$ centered around all lattice points cover the whole space



Distance Function and Covering Radius

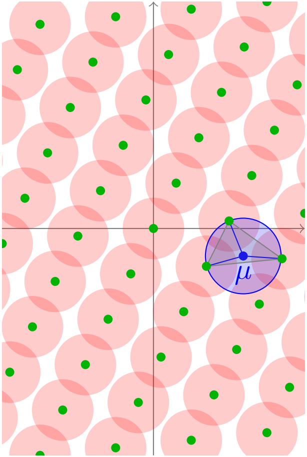
- Distance function

$$\mu(\mathbf{t},\mathcal{L}) = \min_{\mathbf{x}\in\mathcal{L}} \|\mathbf{t}-\mathbf{x}\|$$

- Covering radius

$$\mu(\mathcal{L}) = \max_{\mathbf{t}\in span(\mathcal{L})} \mu(\mathbf{t},\mathcal{L})$$

- Spheres or radius $\mu(\mathcal{L})$ centered around all lattice points cover the whole space



Distance Function and Covering Radius

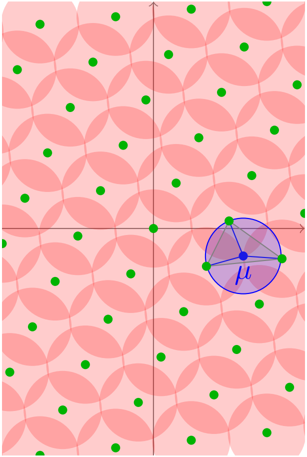
- Distance function

$$\mu(\mathbf{t}, \mathcal{L}) = \min_{\mathbf{x} \in \mathcal{L}} \|\mathbf{t} - \mathbf{x}\|$$

- Covering radius

$$\mu(\mathcal{L}) = \max_{\mathbf{t} \in span(\mathcal{L})} \mu(\mathbf{t}, \mathcal{L})$$

- Spheres or radius $\mu(\mathcal{L})$ centered around all lattice points cover the whole space



Bounding the covering radius

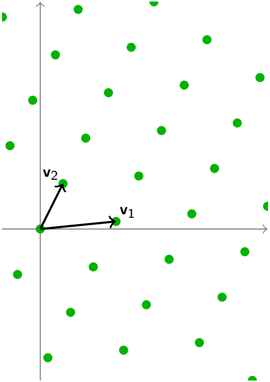
- Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be linearly independent, $\|\mathbf{v}_i\| \leq \lambda_n$
- Tile $\mathbb{R}^n$ with copies of $\mathcal{P}(\mathbf{V}) = \mathbf{V}[0,1)^n$
- If $\mathbf{t} \in \mathbf{x} + \mathcal{P}(\mathbf{V})$, then

$$\|\mathbf{t} - \mathbf{x}\| \leq \sum \|\mathbf{v}_i\| \leq n\lambda_n.$$

- This proves $\mu(\mathcal{L}) \leq n\lambda_n(\mathcal{L})$, and can be further improved:

Theorem

For any lattice $\mathcal{L}$, $\mu(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n(\mathcal{L})$



Bounding the covering radius

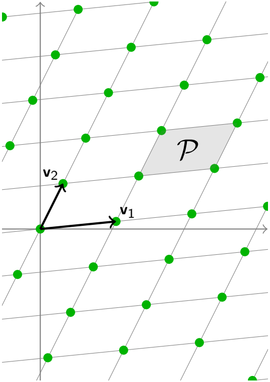
- Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be linearly independent, $\|\mathbf{v}_i\| \leq \lambda_n$
- Tile $\mathbb{R}^n$ with copies of $\mathcal{P}(\mathbf{V}) = \mathbf{V}[0,1)^n$
- If $\mathbf{t} \in \mathbf{x} + \mathcal{P}(\mathbf{V})$, then

$$\|\mathbf{t} - \mathbf{x}\| \leq \sum \|\mathbf{v}_i\| \leq n\lambda_n.$$

- This proves $\mu(\mathcal{L}) \leq n\lambda_n(\mathcal{L})$, and can be further improved:

Theorem

For any lattice $\mathcal{L}$, $\mu(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n(\mathcal{L})$



Bounding the covering radius

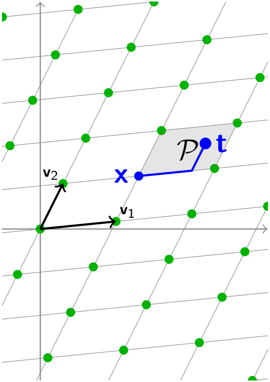
- Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be linearly independent, $\|\mathbf{v}_i\| \leq \lambda_n$
- Tile $\mathbb{R}^n$ with copies of $\mathcal{P}(\mathbf{V}) = \mathbf{V}[0,1)^n$
- If $\mathbf{t} \in \mathbf{x} + \mathcal{P}(\mathbf{V})$, then

$$\|\mathbf{t} - \mathbf{x}\| \leq \sum \|\mathbf{v}_i\| \leq n\lambda_n.$$

- This proves $\mu(\mathcal{L}) \leq n\lambda_n(\mathcal{L})$, and can be further improved:

Theorem

For any lattice $\mathcal{L}$, $\mu(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n(\mathcal{L})$

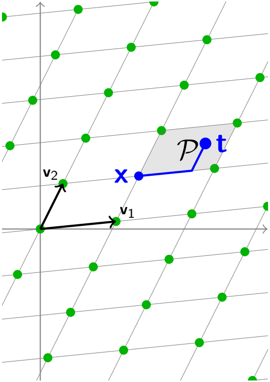


Bounding the covering radius

- Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be linearly independent, $\|\mathbf{v}_i\| \leq \lambda_n$
- Tile $\mathbb{R}^n$ with copies of $\mathcal{P}(\mathbf{V}) = \mathbf{V}[0,1)^n$
- If $\mathbf{t} \in \mathbf{x} + \mathcal{P}(\mathbf{V})$, then

$$\|\mathbf{t} - \mathbf{x}\| \leq \sum \|\mathbf{v}_i\| \leq n\lambda_n.$$

- This proves $\mu(\mathcal{L}) \leq n\lambda_n(\mathcal{L})$, and can be further improved:



Theorem

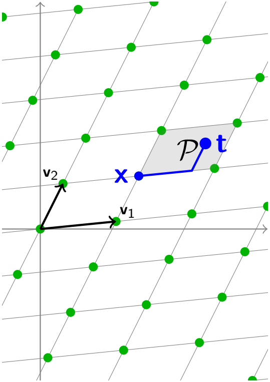
For any lattice $\mathcal{L}$, $\mu(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n(\mathcal{L})$

Bounding the covering radius

- Let $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ be linearly independent, $\|\mathbf{v}_i\| \leq \lambda_n$
- Tile $\mathbb{R}^n$ with copies of $\mathcal{P}(\mathbf{V}) = \mathbf{V}[0,1)^n$
- If $\mathbf{t} \in \mathbf{x} + \mathcal{P}(\mathbf{V})$, then

$$\|\mathbf{t} - \mathbf{x}\| \leq \sum \|\mathbf{v}_i\| \leq n\lambda_n.$$

- This proves $\mu(\mathcal{L}) \leq n\lambda_n(\mathcal{L})$, and can be further improved:



Theorem

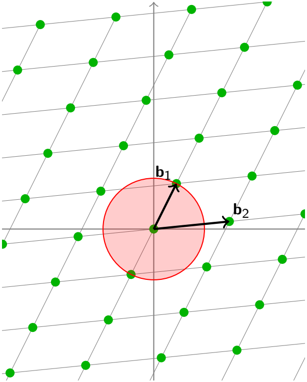
For any lattice $\mathcal{L}$, $\mu(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n(\mathcal{L})$

Bounding the successive minima

- Let $\|\mathbf{b}_1\| = \lambda_1(\mathcal{L})$
- Let $\mathbf{t} = \frac{1}{2}\mathbf{b}_1$
- Then $\mu(\mathbf{t}, \mathcal{L}) \geq \lambda_1/2$
- This proves $\lambda_1(\mathcal{L}) \leq 2\mu(\mathcal{L})$, and can be further improved:

Theorem

For any lattice $\mathcal{L}$, $\lambda_n(\mathcal{L}) \leq 2\mu(\mathcal{L})$

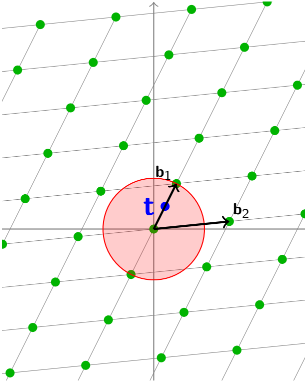


Bounding the successive minima

- Let $\|\mathbf{b}_1\| = \lambda_1(\mathcal{L})$
- Let $\mathbf{t} = \frac{1}{2}\mathbf{b}_1$
- Then $\mu(\mathbf{t},\mathcal{L}) \geq \lambda_1/2$
- This proves $\lambda_1(\mathcal{L}) \leq 2\mu(\mathcal{L})$, and can be further improved:

Theorem

For any lattice $\mathcal{L}$, $\lambda_n(\mathcal{L}) \leq 2\mu(\mathcal{L})$

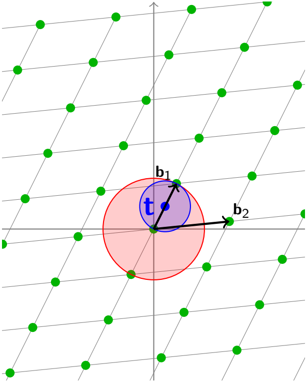


Bounding the successive minima

- Let $\|\mathbf{b}_1\| = \lambda_1(\mathcal{L})$
- Let $\mathbf{t} = \frac{1}{2}\mathbf{b}_1$
- Then $\mu(\mathbf{t}, \mathcal{L}) \geq \lambda_1/2$
- This proves $\lambda_1(\mathcal{L}) \leq 2\mu(\mathcal{L})$, and can be further improved:

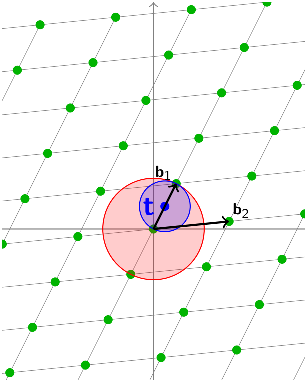
Theorem

For any lattice $\mathcal{L}$, $\lambda_n(\mathcal{L}) \leq 2\mu(\mathcal{L})$



Bounding the successive minima

- Let $\|\mathbf{b}_1\| = \lambda_1(\mathcal{L})$
- Let $\mathbf{t} = \frac{1}{2}\mathbf{b}_1$
- Then $\mu(\mathbf{t}, \mathcal{L}) \geq \lambda_1/2$
- This proves $\lambda_1(\mathcal{L}) \leq 2\mu(\mathcal{L})$, and can be further improved:

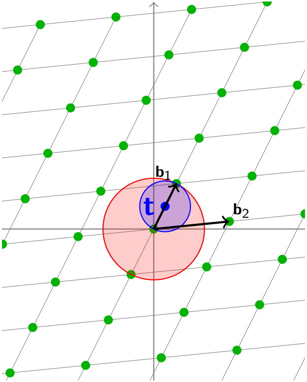


Theorem

For any lattice $\mathcal{L}$, $\lambda_n(\mathcal{L}) \leq 2\mu(\mathcal{L})$

Bounding the successive minima

- Let $\|\mathbf{b}_1\| = \lambda_1(\mathcal{L})$
- Let $\mathbf{t} = \frac{1}{2}\mathbf{b}_1$
- Then $\mu(\mathbf{t}, \mathcal{L}) \geq \lambda_1/2$
- This proves $\lambda_1(\mathcal{L}) \leq 2\mu(\mathcal{L})$, and can be further improved:



Theorem

For any lattice $\mathcal{L}$, $\lambda_n(\mathcal{L}) \leq 2\mu(\mathcal{L})$

Relations among lattice parameters

Theorem

For any lattice $\mathcal{L}$, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2\mu \leq \sqrt{n}\lambda_n$

Remarks:

1. $\mu \approx \lambda_n$ (up to $\sqrt{n}$ factors)
2. For some lattices $\lambda_1 \ll \lambda_2 \ll \ldots \ll \lambda_n$
3. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $2\mu = \sqrt{n}\lambda_n$
4. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $\mu \leq 2\lambda_n$

Problem

Give an explicit construction of a lattice satisfying $\mu \leq 2\lambda_1$

Relations among lattice parameters

Theorem

For any lattice $\mathcal{L}$, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2\mu \leq \sqrt{n}\lambda_n$

Remarks:

1. $\mu \approx \lambda_n$ (up to $\sqrt{n}$ factors)
2. For some lattices $\lambda_1 \ll \lambda_2 \ll \ldots \ll \lambda_n$
3. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $2\mu = \sqrt{n}\lambda_n$
4. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $\mu \leq 2\lambda_n$

Problem

Give an explicit construction of a lattice satisfying $\mu \leq 2\lambda_1$

Relations among lattice parameters

Theorem

For any lattice $\mathcal{L}$, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2\mu \leq \sqrt{n}\lambda_n$

Remarks:

1. $\mu \approx \lambda_n$ (up to $\sqrt{n}$ factors)
2. For some lattices $\lambda_1 \ll \lambda_2 \ll \ldots \ll \lambda_n$
3. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $2\mu = \sqrt{n}\lambda_n$
4. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $\mu \leq 2\lambda_n$

Problem

Give an explicit construction of a lattice satisfying $\mu \leq 2\lambda_1$

Relations among lattice parameters

Theorem

For any lattice $\mathcal{L}$, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2\mu \leq \sqrt{n}\lambda_n$

Remarks:

1. $\mu \approx \lambda_n$ (up to $\sqrt{n}$ factors)
2. For some lattices $\lambda_1 \ll \lambda_2 \ll \ldots \ll \lambda_n$
3. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $2\mu = \sqrt{n}\lambda_n$
4. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $\mu \leq 2\lambda_n$

Problem

Give an explicit construction of a lattice satisfying $\mu \leq 2\lambda_1$

Relations among lattice parameters

Theorem

For any lattice $\mathcal{L}$, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2\mu \leq \sqrt{n}\lambda_n$

Remarks:

1. $\mu \approx \lambda_n$ (up to $\sqrt{n}$ factors)
2. For some lattices $\lambda_1 \ll \lambda_2 \ll \ldots \ll \lambda_n$
3. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $2\mu = \sqrt{n}\lambda_n$
4. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $\mu \leq 2\lambda_n$

Problem

Give an explicit construction of a lattice satisfying $\mu \leq 2\lambda_1$

Relations among lattice parameters

Theorem

For any lattice $\mathcal{L}$, $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2\mu \leq \sqrt{n}\lambda_n$

Remarks:

1. $\mu \approx \lambda_n$ (up to $\sqrt{n}$ factors)
2. For some lattices $\lambda_1 \ll \lambda_2 \ll \ldots \ll \lambda_n$
3. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $2\mu = \sqrt{n}\lambda_n$
4. For some lattices $\lambda_1 = \lambda_2 = \ldots = \lambda_n$ and $\mu \leq 2\lambda_n$

Problem

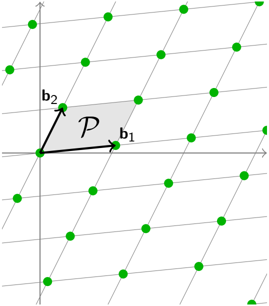
Give an explicit construction of a lattice satisfying $\mu \leq 2\lambda_1$

Determinant

Definition (Determinant)

$\det(\mathcal{L})$ = volume of the fundamental region $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0,1)$

- Different bases define different fundamental regions
- All fundamental regions have the same volume
- The determinant of a lattice can be efficiently computed from any basis.

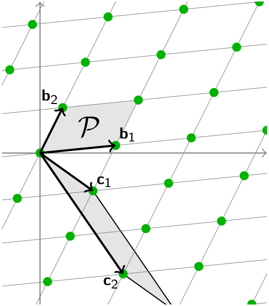


Determinant

Definition (Determinant)

$\det(\mathcal{L})$ = volume of the fundamental region $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0,1)$

- Different bases define different fundamental regions
- All fundamental regions have the same volume
- The determinant of a lattice can be efficiently computed from any basis.

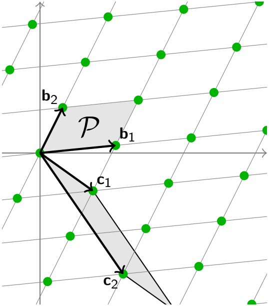


Determinant

Definition (Determinant)

$\det(\mathcal{L})$ = volume of the fundamental region $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0,1)$

- Different bases define different fundamental regions
- All fundamental regions have the same volume
- The determinant of a lattice can be efficiently computed from any basis.

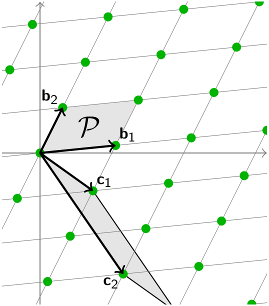


Determinant

Definition (Determinant)

$\det(\mathcal{L})$ = volume of the fundamental region $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0,1)$

- Different bases define different fundamental regions
- All fundamental regions have the same volume
- The determinant of a lattice can be efficiently computed from any basis.



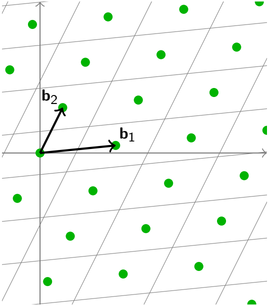
Density estimates

Definition (Centered Fundamental Parallelepiped)

$$\mathcal{P} = \sum_i \mathbf{b}_i \cdot [-1/2, 1/2)$$

- $\mathrm{vol}(\mathcal{P}(\mathbf{B})) = \det(\mathcal{L})$
- $\{\mathbf{x} + \mathcal{P}(\mathbf{B}) \mid \mathbf{x} \in \mathcal{L}\}$ partitions $\mathbb{R}^n$
- For all sufficiently large $S \subseteq \mathbb{R}^n$

$$|S \cap \mathcal{L}| \approx \mathrm{vol}(S)/\det(\mathcal{L})$$



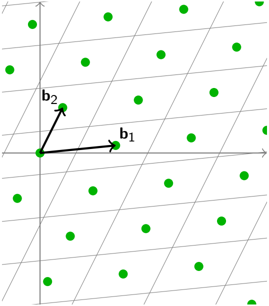
Density estimates

Definition (Centered Fundamental Parallelepiped)

$\mathcal{P} = \sum_i \mathbf{b}_i \cdot [-1/2, 1/2)$

- $\mathrm{vol}(\mathcal{P}(\mathbf{B})) = \det(\mathcal{L})$
- $\{\mathbf{x} + \mathcal{P}(\mathbf{B}) \mid \mathbf{x} \in \mathcal{L}\}$ partitions $\mathbb{R}^n$
- For all sufficiently large $S \subseteq \mathbb{R}^n$

$$|S \cap \mathcal{L}| \approx \mathrm{vol}(S)/\det(\mathcal{L})$$



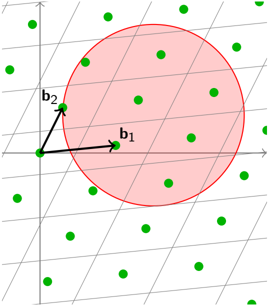
Density estimates

Definition (Centered Fundamental Parallelepiped)

$\mathcal{P} = \sum_i \mathbf{b}_i \cdot [-1/2, 1/2)$

- $\mathrm{vol}(\mathcal{P}(\mathbf{B})) = \det(\mathcal{L})$
- $\{\mathbf{x} + \mathcal{P}(\mathbf{B}) \mid \mathbf{x} \in \mathcal{L}\}$ partitions $\mathbb{R}^n$
- For all sufficiently large $S \subseteq \mathbb{R}^n$

$$|S \cap \mathcal{L}| \approx \mathrm{vol}(S) / \det(\mathcal{L})$$

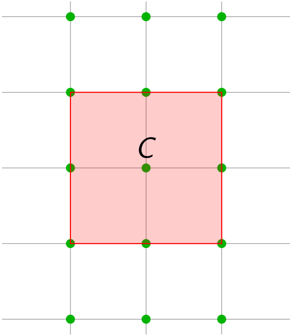


Minkowski's convex body theorem

Theorem (Convex Body)

Let $C \subset \mathbb{R}^n$ be a symmetric convex body. If vol$(C) > 2^n$, then C contains a nonzero integer vector

- $C = \mathbf{B}^{-1}[-r,r]^n$ has volume $\det(\mathbf{B})^{-1}(2r)^n = 2^n$
- C contains $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$
- $\mathbf{B}C = [-r,r]^n$ contains $\mathbf{Bx}$
- $\lambda_1(\mathcal{L}) \le \sqrt{n}r = \sqrt{n}\det(\mathcal{L})^{1/n}$



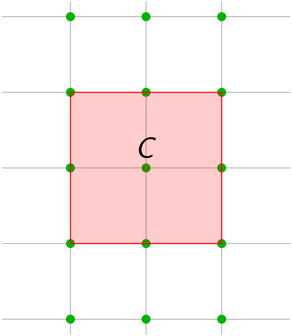
Minkowski's convex body theorem

Theorem (Convex Body)

Let $C \subset \mathbb{R}^n$ be a symmetric convex body. If vol$(C) > 2^n$, then C contains a nonzero integer vector

Let $\mathcal{L} = \mathbf{B}\mathbb{Z}^n$ and $r = \det(\mathcal{L})^{1/n}$. Then,

- $C = \mathbf{B}^{-1}[-r, r]^n$ has volume $\det(\mathbf{B})^{-1}(2r)^n = 2^n$
- C contains $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$
- $\mathbf{B}C = [-r, r]^n$ contains $\mathbf{Bx}$
- $\lambda_1(\mathcal{L}) \leq \sqrt{n}r = \sqrt{n}\det(\mathcal{L})^{1/n}$



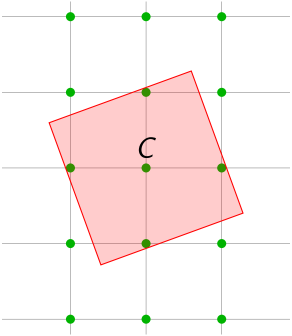
Minkowski's convex body theorem

Theorem (Convex Body)

Let $C \subset \mathbb{R}^n$ be a symmetric convex body. If vol$(C) > 2^n$, then C contains a nonzero integer vector

Let $\mathcal{L} = \mathbf{B}\mathbb{Z}^n$ and $r = \det(\mathcal{L})^{1/n}$. Then,

- $C = \mathbf{B}^{-1}[-r, r]^n$ has volume $\det(\mathbf{B})^{-1}(2r)^n = 2^n$
- C contains $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$
- $\mathbf{B}C = [-r, r]^n$ contains $\mathbf{Bx}$
- $\lambda_1(\mathcal{L}) \leq \sqrt{n}r = \sqrt{n}\det(\mathcal{L})^{1/n}$



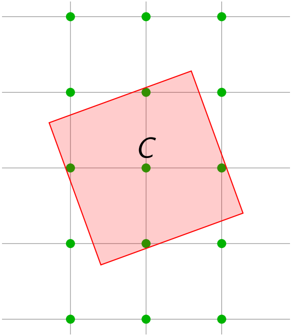
Minkowski's convex body theorem

Theorem (Convex Body)

Let $C \subset \mathbb{R}^n$ be a symmetric convex body. If vol$(C) > 2^n$, then C contains a nonzero integer vector

Let $\mathcal{L} = \mathbf{B}\mathbb{Z}^n$ and $r = \det(\mathcal{L})^{1/n}$. Then,

- $C = \mathbf{B}^{-1}[-r, r]^n$ has volume $\det(\mathbf{B})^{-1}(2r)^n = 2^n$
- C contains $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$
- $\mathbf{B}C = [-r, r]^n$ contains $\mathbf{Bx}$
- $\lambda_1(\mathcal{L}) \leq \sqrt{n}r = \sqrt{n}\det(\mathcal{L})^{1/n}$



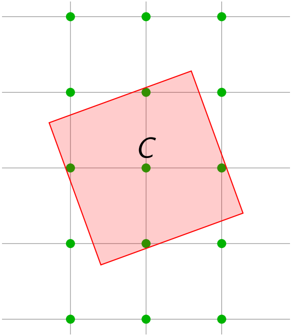
Minkowski's convex body theorem

Theorem (Convex Body)

Let $C \subset \mathbb{R}^n$ be a symmetric convex body. If vol$(C) > 2^n$, then C contains a nonzero integer vector

Let $\mathcal{L} = \mathbf{B}\mathbb{Z}^n$ and $r = \det(\mathcal{L})^{1/n}$. Then,

- $C = \mathbf{B}^{-1}[-r, r]^n$ has volume $\det(\mathbf{B})^{-1}(2r)^n = 2^n$
- C contains $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$
- $\mathbf{B}C = [-r, r]^n$ contains $\mathbf{Bx}$
- $\lambda_1(\mathcal{L}) \leq \sqrt{n}r = \sqrt{n}\det(\mathcal{L})^{1/n}$



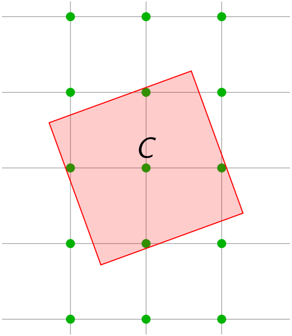
Minkowski's convex body theorem

Theorem (Convex Body)

Let $C \subset \mathbb{R}^n$ be a symmetric convex body. If vol$(C) > 2^n$, then C contains a nonzero integer vector

Let $\mathcal{L} = \mathbf{B}\mathbb{Z}^n$ and $r = \det(\mathcal{L})^{1/n}$. Then,

- $C = \mathbf{B}^{-1}[-r, r]^n$ has volume $\det(\mathbf{B})^{-1}(2r)^n = 2^n$
- C contains $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$
- $\mathbf{B}C = [-r, r]^n$ contains $\mathbf{Bx}$
- $\lambda_1(\mathcal{L}) \leq \sqrt{n}r = \sqrt{n}\det(\mathcal{L})^{1/n}$



Minkowski's second theorem

Theorem (Minkowski)

$$\lambda_1(\mathcal{L}) \leq \left(\prod_i \lambda_i(\mathcal{L})\right)^{1/n} \leq \sqrt{n}\,\det(\mathcal{L})^{1/n}$$

- For $\mathbb{Z}^n$, $\lambda_1 = (\prod_i \lambda_i)^{1/n} = 1$ is smaller than Minkowski's bound by $\sqrt{n}$
- $\lambda_1(\mathcal{L})$ can be arbitrarily smaller than Minkowski's bound
- $(\prod_i \lambda_i(\mathcal{L}))^{1/n}$ is never smaller than Minkowski's bound by more than $\sqrt{n}$
- Can you find lattices with $(\prod_i \lambda_i(\mathcal{L}))^{1/n} \geq \Omega(\sqrt{n}) \det(\mathcal{L})^{1/n}$ within a constant from Minkowski's bound?

Minkowski's second theorem

Theorem (Minkowski)

$$\lambda_1(\mathcal{L}) \leq \left(\prod_i \lambda_i(\mathcal{L})\right)^{1/n} \leq \sqrt{n}\,\det(\mathcal{L})^{1/n}$$

- For $\mathbb{Z}^n$, $\lambda_1 = (\prod_i \lambda_i)^{1/n} = 1$ is smaller than Minkowski's bound by $\sqrt{n}$
- $\lambda_1(\mathcal{L})$ can be arbitrarily smaller than Minkowski's bound
- $(\prod_i \lambda_i(\mathcal{L}))^{1/n}$ is never smaller than Minkowski's bound by more than $\sqrt{n}$
- Can you find lattices with $(\prod_i \lambda_i(\mathcal{L}))^{1/n} \geq \Omega(\sqrt{n})\det(\mathcal{L})^{1/n}$ within a constant from Minkowski's bound?

Minkowski's second theorem

Theorem (Minkowski)

$$\lambda_1(\mathcal{L}) \leq \left(\prod_i \lambda_i(\mathcal{L})\right)^{1/n} \leq \sqrt{n}\,\det(\mathcal{L})^{1/n}$$

- For $\mathbb{Z}^n$, $\lambda_1 = (\prod_i \lambda_i)^{1/n} = 1$ is smaller than Minkowski's bound by $\sqrt{n}$
- $\lambda_1(\mathcal{L})$ can be arbitrarily smaller than Minkowski's bound
- $(\prod_i \lambda_i(\mathcal{L}))^{1/n}$ is never smaller than Minkowski's bound by more than $\sqrt{n}$
- Can you find lattices with $(\prod_i \lambda_i(\mathcal{L}))^{1/n} \geq \Omega(\sqrt{n})\det(\mathcal{L})^{1/n}$ within a constant from Minkowski's bound?

Minkowski's second theorem

Theorem (Minkowski)

$$\lambda_1(\mathcal{L}) \leq \left(\prod_i \lambda_i(\mathcal{L})\right)^{1/n} \leq \sqrt{n}\,\det(\mathcal{L})^{1/n}$$

- For $\mathbb{Z}^n$, $\lambda_1 = (\prod_i \lambda_i)^{1/n} = 1$ is smaller than Minkowski's bound by $\sqrt{n}$
- $\lambda_1(\mathcal{L})$ can be arbitrarily smaller than Minkowski's bound
- $(\prod_i \lambda_i(\mathcal{L}))^{1/n}$ is never smaller than Minkowski's bound by more than $\sqrt{n}$
- Can you find lattices with $(\prod_i \lambda_i(\mathcal{L}))^{1/n} \geq \Omega(\sqrt{n})\det(\mathcal{L})^{1/n}$ within a constant from Minkowski's bound?

Minkowski's second theorem

Theorem (Minkowski)

$$\lambda_1(\mathcal{L}) \leq \left(\prod_i \lambda_i(\mathcal{L})\right)^{1/n} \leq \sqrt{n}\det(\mathcal{L})^{1/n}$$

- For $\mathbb{Z}^n$, $\lambda_1 = (\prod_i \lambda_i)^{1/n} = 1$ is smaller than Minkowski's bound by $\sqrt{n}$
- $\lambda_1(\mathcal{L})$ can be arbitrarily smaller than Minkowski's bound
- $(\prod_i \lambda_i(\mathcal{L}))^{1/n}$ is never smaller than Minkowski's bound by more than $\sqrt{n}$
- Can you find lattices with $(\prod_i \lambda_i(\mathcal{L}))^{1/n} \geq \Omega(\sqrt{n})\det(\mathcal{L})^{1/n}$ within a constant from Minkowski's bound?

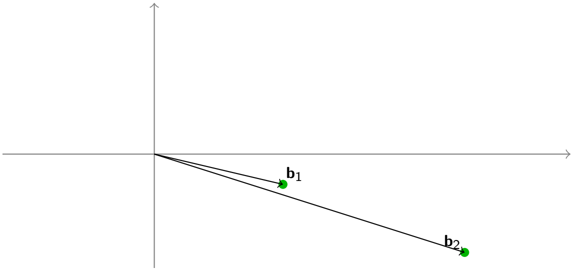
Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

Shortest Vector Problem

Definition (Shortest Vector Problem, SVP)

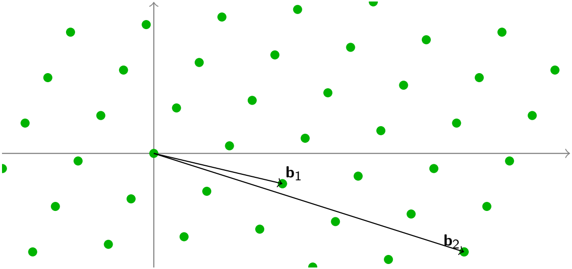
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \lambda_1$



Shortest Vector Problem

Definition (Shortest Vector Problem, SVP)

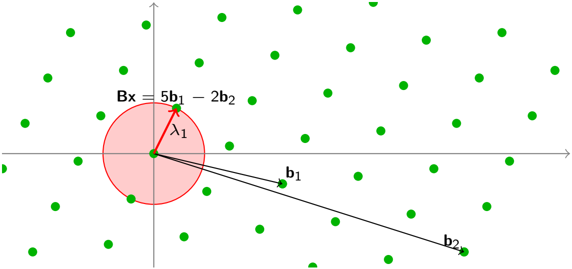
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \le \lambda_1$



Shortest Vector Problem

Definition (Shortest Vector Problem, SVP)

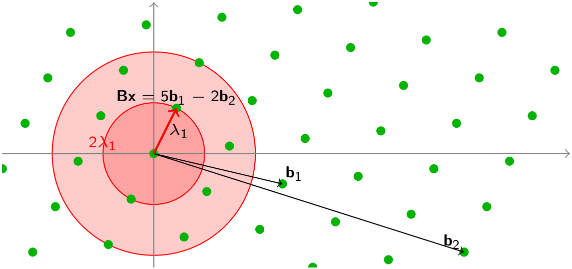
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \lambda_1$



Shortest Vector Problem

Definition (Shortest Vector Problem, SVP_γ)

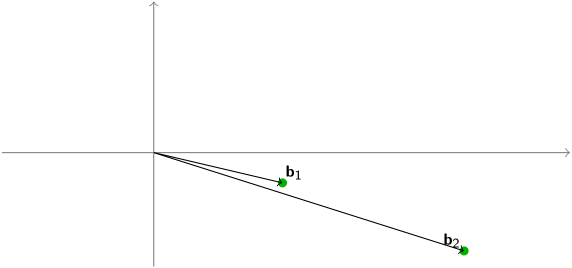
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \gamma\lambda_1$



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

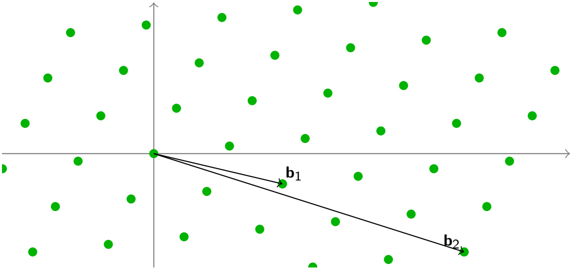
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \lambda_n$



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

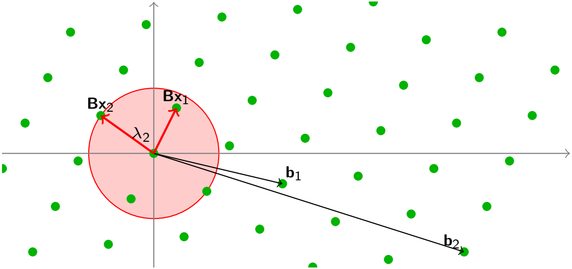
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{B}\mathbf{x}_1, \ldots, \mathbf{B}\mathbf{x}_n$ of length (at most) $\max_i \|\mathbf{B}\mathbf{x}_i\| \leq \lambda_n$



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

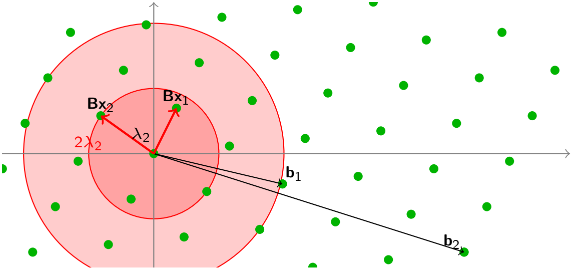
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \lambda_n$



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP_γ)

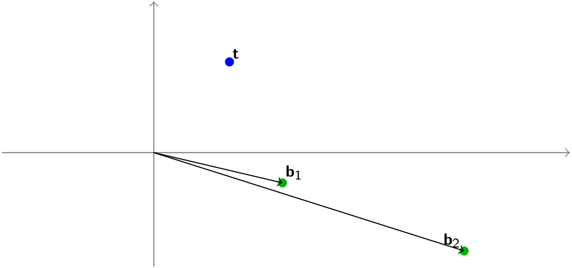
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \gamma\lambda_n$



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

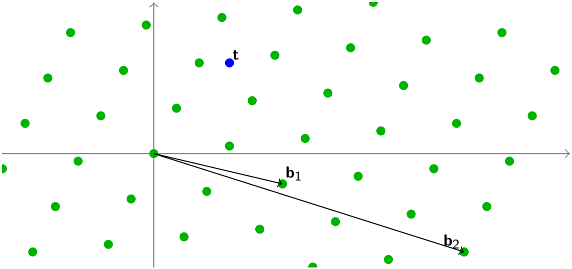
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \mu$ from the target



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

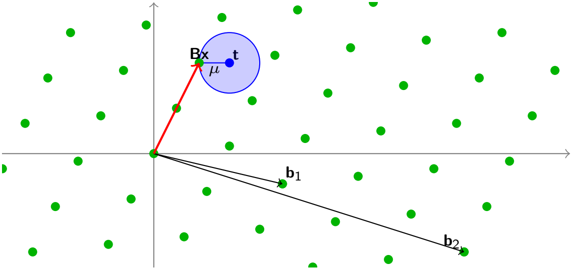
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx}-\mathbf{t}\| \leq \mu$ from the target



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

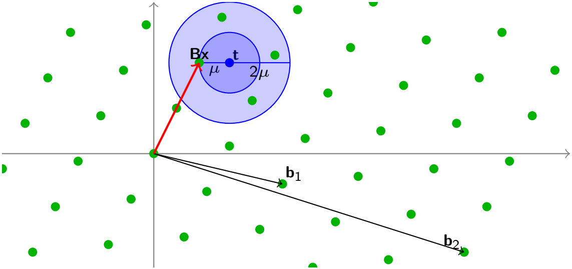
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx}-\mathbf{t}\| \leq \mu$ from the target



Closest Vector Problem

Definition (Closest Vector Problem, CVP$_\gamma$)

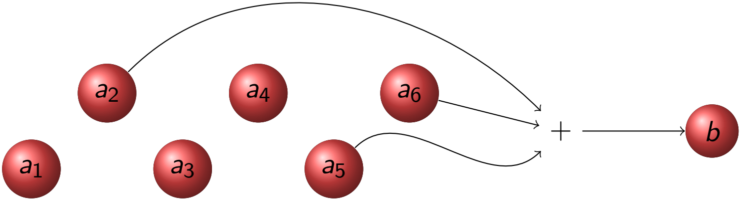
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \gamma\mu$ from the target



NP-hardness of CVP

Definition (Subset Sum)

Given $a_1, \ldots, a_n, b \in \mathbb{Z}$ find $S \subseteq \{1, \ldots, n\}$ s.t. $\sum_{i \in S} a_i = b$



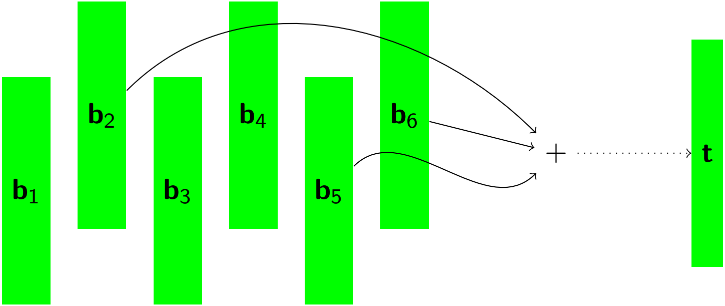
Theorem

$\|\mathbf{Bx} - \mathbf{t}\| \leq \sqrt{n}$ *if and only if* $\mathbf{x} \in \{0,1\}^n$ *and* $\sum_{x_i=1} a_i = b$.

NP-hardness of CVP

Definition (Subset Sum)

Given $a_1, \ldots, a_n, b \in \mathbb{Z}$ find $S \subseteq \{1, \ldots, n\}$ s.t. $\sum_{i \in S} a_i = b$



Theorem

$\|\mathbf{Bx} - \mathbf{t}\| \leq \sqrt{n}$ if and only if $\mathbf{x} \in \{0,1\}^n$ and $\sum_{x_i=1} a_i = b$.

NP-hardness of CVP

Definition (Subset Sum)

Given $a_1, \ldots, a_n, b \in \mathbb{Z}$ find $S \subseteq \{1, \ldots, n\}$ s.t. $\sum_{i \in S} a_i = b$

$$\mathbf{B} = \left[\begin{array}{c|c|c} a_1 & \cdots & a_n \\ 2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 2 \end{array}\right] \qquad \mathbf{t} = \begin{bmatrix} b \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad \mathbf{Bx}-\mathbf{t} = \begin{bmatrix} \sum_i a_i x_i - b \\ 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \end{bmatrix}$$

Theorem

$\|\mathbf{Bx} - \mathbf{t}\| \leq \sqrt{n}$ *if and only if* $\mathbf{x} \in \{0,1\}^n$ *and* $\sum_{x_i=1} a_i = b$.

NP-hardness of CVP

Definition (Subset Sum)

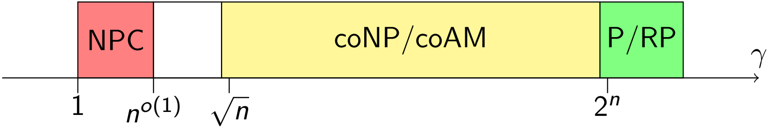
Given $a_1, \ldots, a_n, b \in \mathbb{Z}$ find $S \subseteq \{1, \ldots, n\}$ s.t. $\sum_{i \in S} a_i = b$

$$\mathbf{B} = \left[\begin{array}{c|c|c} a_1 & \cdots & a_n \\ 2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 2 \end{array}\right] \qquad \mathbf{t} = \begin{bmatrix} b \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad \mathbf{Bx} - \mathbf{t} = \begin{bmatrix} \sum_i a_i x_i - b \\ 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \end{bmatrix}$$

Theorem

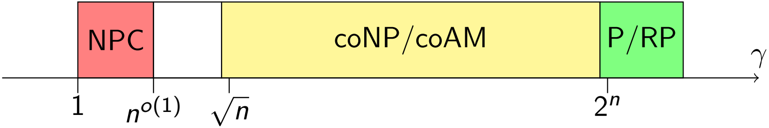
$\|\mathbf{Bx} - \mathbf{t}\| \le \sqrt{n}$ *if and only if* $\mathbf{x} \in \{0, 1\}^n$ *and* $\sum_{x_i = 1} a_i = b$.

Complexity of CVP, SVP, SIVP



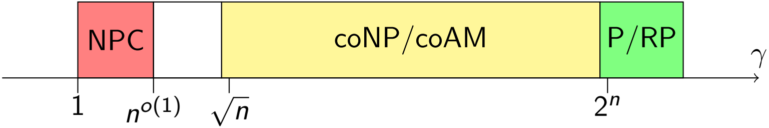
- Best algorithm for exact solution takes time 2^n [MV10]
- (Almost) NP-hard for factors up to $\gamma = n^{1/loglogn}$. [Ajtai96,. . . ,HR07]
- Polynomial time for slightly subexponential γ [Schnorr93+AKS01,GN08+MV10]
- Unlikely to be NP-hard for $\gamma \geq \sqrt{n/\log n}$ [GG01,AR04]

Complexity of CVP, SVP, SIVP



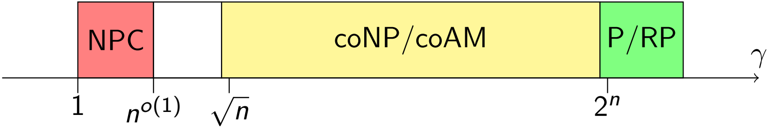
- Best algorithm for exact solution takes time 2^n [MV10]
- (Almost) NP-hard for factors up to $\gamma = n^{1/loglogn}$. [Ajtai96,...,HR07]
- Polynomial time for slightly subexponential γ [Schnorr93+AKS01,GN08+MV10]
- Unlikely to be NP-hard for $\gamma \geq \sqrt{n/\log n}$ [GG01,AR04]

Complexity of CVP, SVP, SIVP



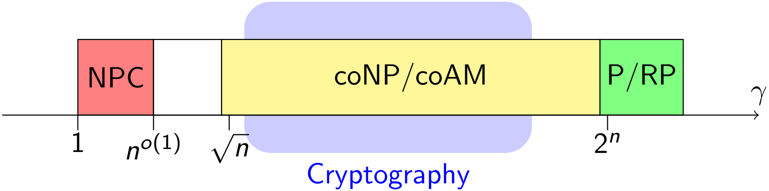
- Best algorithm for exact solution takes time 2^n [MV10]
- (Almost) NP-hard for factors up to $\gamma = n^{1/loglogn}$. [Ajtai96,. . . ,HR07]
- Polynomial time for slightly subexponential γ [Schnorr93+AKS01,GN08+MV10]
- Unlikely to be NP-hard for $\gamma \geq \sqrt{n/\log n}$ [GG01,AR04]

Complexity of CVP, SVP, SIVP



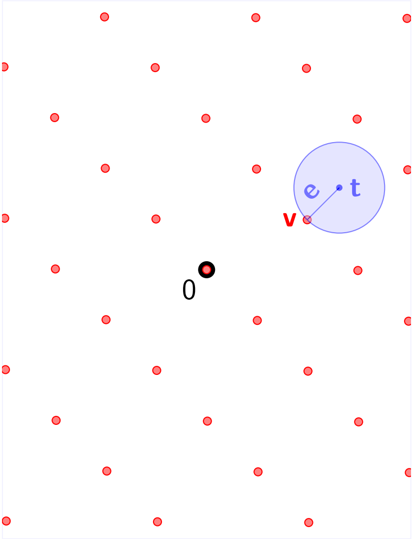
- Best algorithm for exact solution takes time 2^n [MV10]
- (Almost) NP-hard for factors up to $\gamma = n^{1/loglogn}$. [Ajtai96,...,HR07]
- Polynomial time for slightly subexponential γ [Schnorr93+AKS01,GN08+MV10]
- Unlikely to be NP-hard for $\gamma \geq \sqrt{n/\log n}$ [GG01,AR04]

Complexity of CVP, SVP, SIVP



- Best algorithm for exact solution takes time 2^n [MV10]
- (Almost) NP-hard for factors up to $\gamma = n^{1/loglogn}$. [Ajtai96,. . . ,HR07]
- Polynomial time for slightly subexponential γ [Schnorr93+AKS01,GN08+MV10]
- Unlikely to be NP-hard for $\gamma \geq \sqrt{n/\log n}$ [GG01,AR04]

CVP and lattice cosets

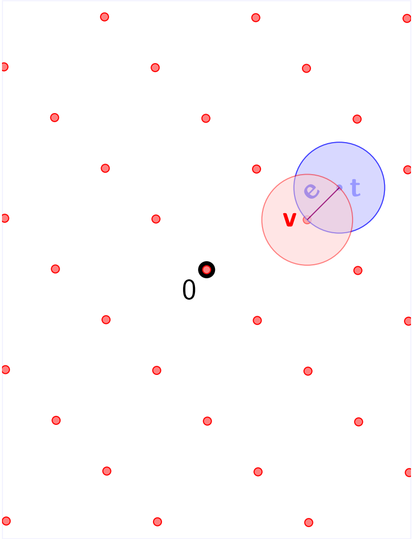


- Lattice Λ, target $\mathbf{t}$
- CVP: Find $\mathbf{v}$ such that $\mathbf{e} = \mathbf{t} - \mathbf{v}$ is shortest possible
- $\mathbf{t}' = \mathbf{t} + \mathbf{Bx}$
- $\mathbf{v} = \mathbf{v}' - \mathbf{Bx}$

Definition (Coset CVP)

Given a lattice coset $\mathbf{t} + \mathcal{L}$, find the (approximately) shortest element of $\mathbf{t} + \mathcal{L}$.

CVP and lattice cosets

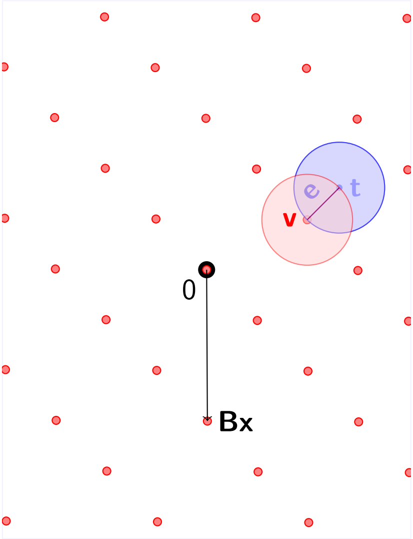


- Lattice Λ, target $\mathbf{t}$
- CVP: Find $\mathbf{v}$ such that $\mathbf{e} = \mathbf{t} - \mathbf{v}$ is shortest possible
- $\mathbf{t}' = \mathbf{t} + \mathbf{Bx}$
- $\mathbf{v} = \mathbf{v}' - \mathbf{Bx}$

Definition (Coset CVP)

Given a lattice coset $\mathbf{t} + \mathcal{L}$, find the (approximately) shortest element of $\mathbf{t} + \mathcal{L}$.

CVP and lattice cosets

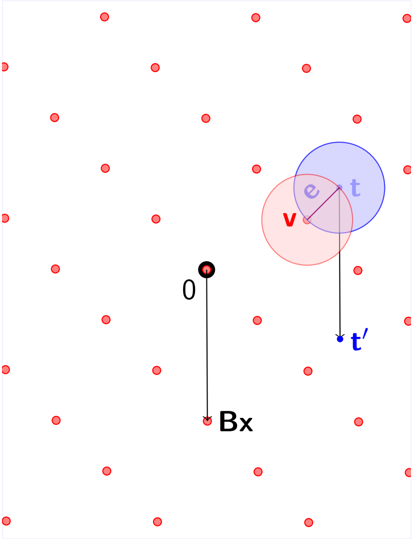


- Lattice Λ, target $\mathbf{t}$
- CVP: Find $\mathbf{v}$ such that $\mathbf{e} = \mathbf{t} - \mathbf{v}$ is shortest possible
- $\mathbf{t}' = \mathbf{t} + \mathbf{Bx}$
- $\mathbf{v} = \mathbf{v}' - \mathbf{Bx}$

Definition (Coset CVP)

Given a lattice coset $\mathbf{t} + \mathcal{L}$, find the (approximately) shortest element of $\mathbf{t} + \mathcal{L}$.

CVP and lattice cosets

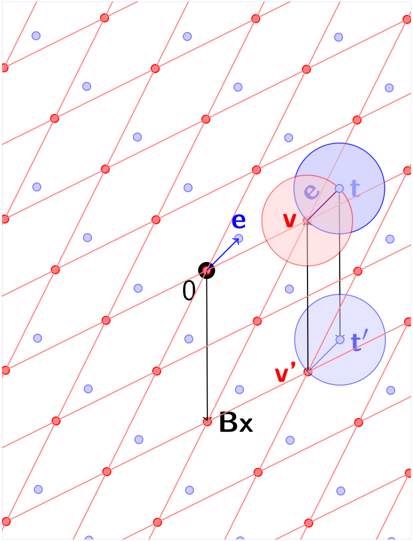


- Lattice Λ, target $\mathbf{t}$
- CVP: Find $\mathbf{v}$ such that $\mathbf{e} = \mathbf{t} - \mathbf{v}$ is shortest possible
- $\mathbf{t}' = \mathbf{t} + \mathbf{Bx}$
- $\mathbf{v} = \mathbf{v}' - \mathbf{Bx}$

Definition (Coset CVP)

Given a lattice coset $\mathbf{t} + \mathcal{L}$, find the (approximately) shortest element of $\mathbf{t} + \mathcal{L}$.

CVP and lattice cosets

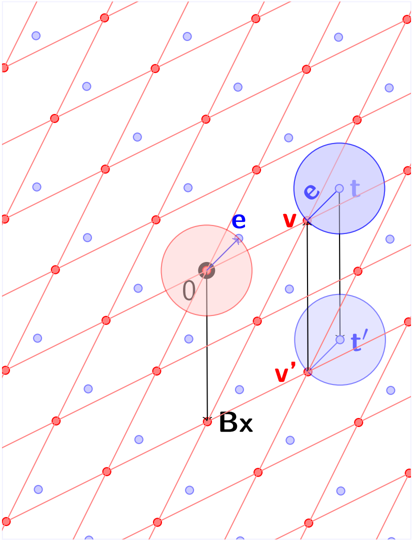


- Lattice Λ, target $\mathbf{t}$
- CVP: Find $\mathbf{v}$ such that $\mathbf{e} = \mathbf{t} - \mathbf{v}$ is shortest possible
- $\mathbf{t}' = \mathbf{t} + \mathbf{Bx}$
- $\mathbf{v} = \mathbf{v}' - \mathbf{Bx}$

Definition (Coset CVP)

Given a lattice coset $\mathbf{t} + \mathcal{L}$, find the (approximately) shortest element of $\mathbf{t} + \mathcal{L}$.

CVP and lattice cosets



- Lattice Λ, target $\mathbf{t}$
- CVP: Find $\mathbf{v}$ such that $\mathbf{e} = \mathbf{t} - \mathbf{v}$ is shortest possible
- $\mathbf{t}' = \mathbf{t} + \mathbf{Bx}$
- $\mathbf{v} = \mathbf{v}' - \mathbf{Bx}$

Definition (Coset CVP)

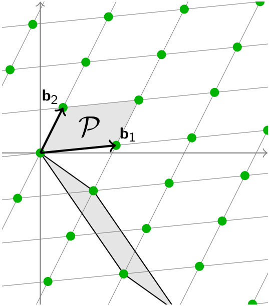
Given a lattice coset $\mathbf{t} + \mathcal{L}$, find the (approximately) shortest element of $\mathbf{t} + \mathcal{L}$.

Working modulo a lattice

Definition (Fundamental Region)

$D \subset \mathbb{R}^n$ is a fundamental region for $\mathcal{L}$ if $\{D + \mathbf{x} \mid \mathbf{x} \in \mathcal{L}\}$ is a partition of $\mathbb{R}^n$.

- $(\mathcal{L}, +)$ is a subgroup of $(\mathbb{R}^n, +)$
- One can form the quotien group $\mathbb{R}^n/\mathcal{L}$
- Elements of $\mathbb{R}^n/\mathcal{L}$ are cosets $\mathbf{t} + \mathcal{L}$
- Any fundamental region D gives a set of standard representatives
- $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0, 1) \equiv \mathbb{R}^n/\mathcal{L}$

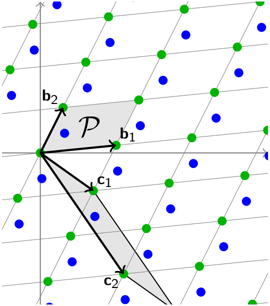


Working modulo a lattice

Definition (Fundamental Region)

$D \subset \mathbb{R}^n$ is a fundamental region for $\mathcal{L}$ if $\{D + \mathbf{x} \mid \mathbf{x} \in \mathcal{L}\}$ is a partition of $\mathbb{R}^n$.

- $(\mathcal{L}, +)$ is a subgroup of $(\mathbb{R}^n, +)$
- One can form the quotien group $\mathbb{R}^n/\mathcal{L}$
- Elements of $\mathbb{R}^n/\mathcal{L}$ are cosets $\mathbf{t} + \mathcal{L}$
- Any fundamental region D gives a set of standard representatives
- $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0, 1) \equiv \mathbb{R}^n/\mathcal{L}$

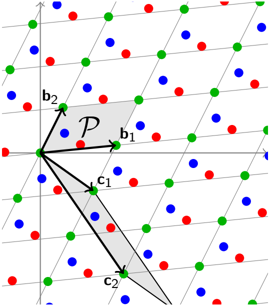


Working modulo a lattice

Definition (Fundamental Region)

$D \subset \mathbb{R}^n$ is a fundamental region for $\mathcal{L}$ if $\{D + \mathbf{x} \mid \mathbf{x} \in \mathcal{L}\}$ is a partition of $\mathbb{R}^n$.

- $(\mathcal{L}, +)$ is a subgroup of $(\mathbb{R}^n, +)$
- One can form the quotien group $\mathbb{R}^n/\mathcal{L}$
- Elements of $\mathbb{R}^n/\mathcal{L}$ are cosets $\mathbf{t} + \mathcal{L}$
- Any fundamental region D gives a set of standard representatives
- $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0, 1) \equiv \mathbb{R}^n/\mathcal{L}$

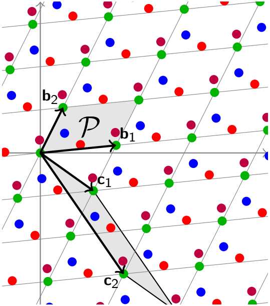


Working modulo a lattice

Definition (Fundamental Region)

$D \subset \mathbb{R}^n$ is a fundamental region for $\mathcal{L}$ if $\{D + \mathbf{x} \mid \mathbf{x} \in \mathcal{L}\}$ is a partition of $\mathbb{R}^n$.

- $(\mathcal{L}, +)$ is a subgroup of $(\mathbb{R}^n, +)$
- One can form the quotien group $\mathbb{R}^n/\mathcal{L}$
- Elements of $\mathbb{R}^n/\mathcal{L}$ are cosets $\mathbf{t} + \mathcal{L}$
- Any fundamental region D gives a set of standard representatives
- $\mathcal{P} = \sum_i \mathbf{b}_i \cdot [0,1) \equiv \mathbb{R}^n/\mathcal{L}$



Interlude: CVP One-way Function?

Candidate OWF

Key: a hard lattice $\mathcal{L}$

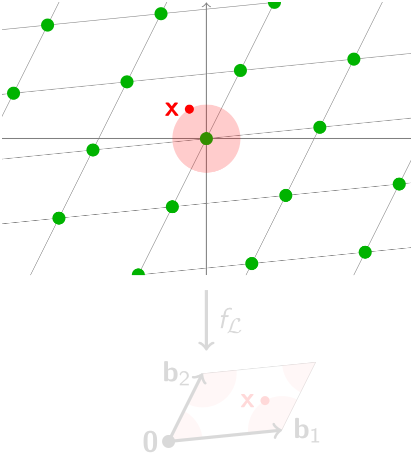
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$

Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $g_{\mathcal{L}}(\mathbf{x})$ is almost uniform

Question

Is $f_{\mathcal{L}}$ hard to invert?



Interlude: CVP One-way Function?

Candidate OWF

Key: a hard lattice $\mathcal{L}$

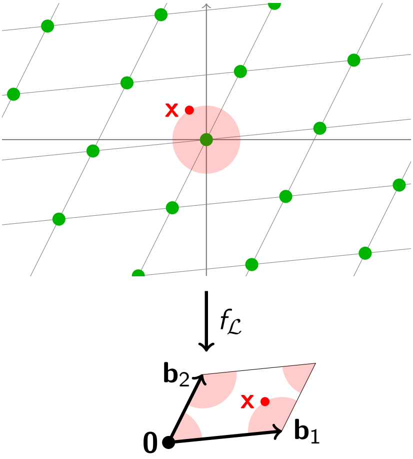
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$

Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $g_{\mathcal{L}}(\mathbf{x})$ is almost uniform

Question

Is $f_{\mathcal{L}}$ hard to invert?



Interlude: CVP One-way Function?

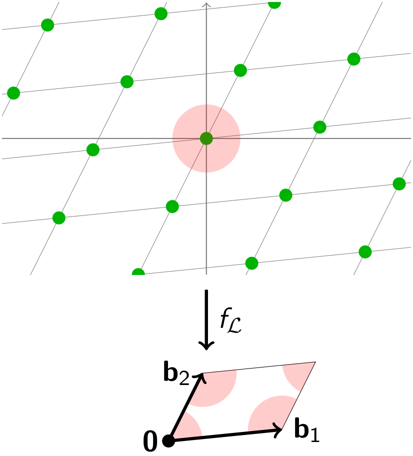
Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $g_{\mathcal{L}}(\mathbf{x})$ is almost uniform

Question

Is $f_{\mathcal{L}}$ hard to invert?



Interlude: CVP One-way Function?

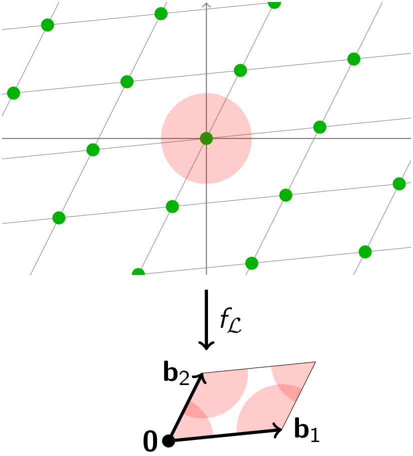
Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $g_{\mathcal{L}}(\mathbf{x})$ is almost uniform

Question

Is $f_{\mathcal{L}}$ hard to invert?



Interlude: CVP One-way Function?

Candidate OWF

Key: a hard lattice $\mathcal{L}$

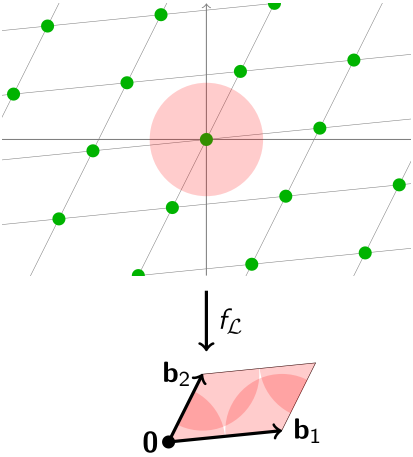
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$

Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $g_{\mathcal{L}}(\mathbf{x})$ is almost uniform

Question

Is $f_{\mathcal{L}}$ hard to invert?



Interlude: CVP One-way Function?

Candidate OWF

Key: a hard lattice $\mathcal{L}$

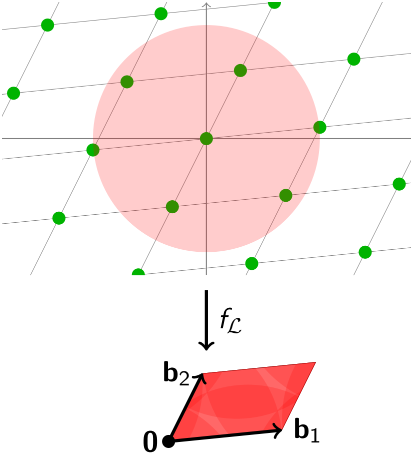
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$

Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $g_{\mathcal{L}}(\mathbf{x})$ is almost uniform

Question

Is $f_{\mathcal{L}}$ hard to invert?



Interlude: CVP One-way Function?

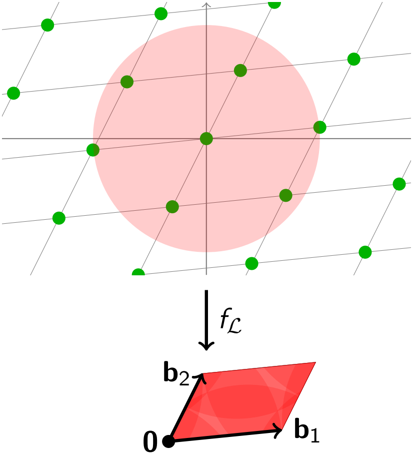
Candidate OWF

Key: a hard lattice $\mathcal{L}$

Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$

Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $g_{\mathcal{L}}(\mathbf{x})$ is almost uniform



Question

Is $f_{\mathcal{L}}$ hard to invert?

Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

The Dual

- A vector space over $\mathbb{R}$ is a set of vectors V with
  - a vector addition operation $\mathbf{x} + \mathbf{y} \in V$
  - a scalar multiplication $a \cdot \mathbf{x} \in V$
- The dual of a vector space V is the set $V^* = Hom(V, \mathbb{R})$ of linear functions $\phi : V \to \mathbb{R}$, typically represented as vectors $\mathbf{x} \in V$, where $\phi_{\mathbf{x}}(\mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$
- The dual of a lattice Λ is defined similarly as the set of linear functions $\phi_{\mathbf{x}} : \Lambda \to \mathbb{Z}$ represented as vectors $\mathbf{x} \in span(\Lambda)$.

Definition (Dual lattice)

The dual of a lattice Λ is the set of all vectors $\mathbf{x} \in span(\Lambda)$ such that $\langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}$ for all $\mathbf{v} \in \Lambda$

The Dual

- A vector space over $\mathbb{R}$ is a set of vectors V with
  - a vector addition operation $\mathbf{x} + \mathbf{y} \in V$
  - a scalar multiplication $a \cdot \mathbf{x} \in V$
- The dual of a vector space V is the set $V^* = Hom(V, \mathbb{R})$ of linear functions $\phi : V \to \mathbb{R}$, typically represented as vectors $\mathbf{x} \in V$, where $\phi_{\mathbf{x}}(\mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$
- The dual of a lattice Λ is defined similarly as the set of linear functions $\phi_{\mathbf{x}} : \Lambda \to \mathbb{Z}$ represented as vectors $\mathbf{x} \in span(\Lambda)$.

Definition (Dual lattice)

The dual of a lattice Λ is the set of all vectors $\mathbf{x} \in span(\Lambda)$ such that $\langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}$ for all $\mathbf{v} \in \Lambda$

The Dual

- A vector space over $\mathbb{R}$ is a set of vectors V with
  - a vector addition operation $\mathbf{x} + \mathbf{y} \in V$
  - a scalar multiplication $a \cdot \mathbf{x} \in V$
- The dual of a vector space V is the set $V^* = Hom(V, \mathbb{R})$ of linear functions $\phi : V \to \mathbb{R}$, typically represented as vectors $\mathbf{x} \in V$, where $\phi_{\mathbf{x}}(\mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$
- The dual of a lattice Λ is defined similarly as the set of linear functions $\phi_{\mathbf{x}} : \Lambda \to \mathbb{Z}$ represented as vectors $\mathbf{x} \in span(\Lambda)$.

Definition (Dual lattice)

The dual of a lattice Λ is the set of all vectors $\mathbf{x} \in span(\Lambda)$ such that $\langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}$ for all $\mathbf{v} \in \Lambda$

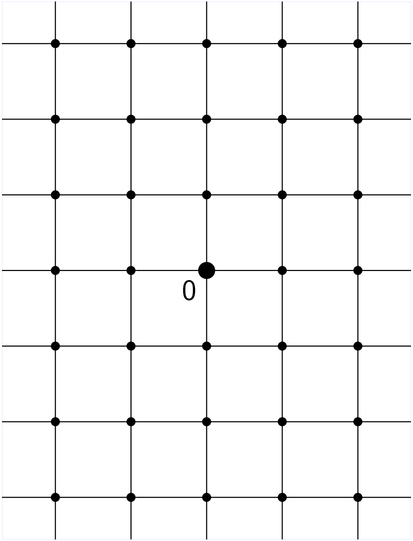
The Dual

- A vector space over $\mathbb{R}$ is a set of vectors V with
  - a vector addition operation $\mathbf{x} + \mathbf{y} \in V$
  - a scalar multiplication $a \cdot \mathbf{x} \in V$
- The dual of a vector space V is the set $V^* = Hom(V, \mathbb{R})$ of linear functions $\phi : V \to \mathbb{R}$, typically represented as vectors $\mathbf{x} \in V$, where $\phi_{\mathbf{x}}(\mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$
- The dual of a lattice Λ is defined similarly as the set of linear functions $\phi_{\mathbf{x}} \colon \Lambda \to \mathbb{Z}$ represented as vectors $\mathbf{x} \in span(\Lambda)$.

Definition (Dual lattice)

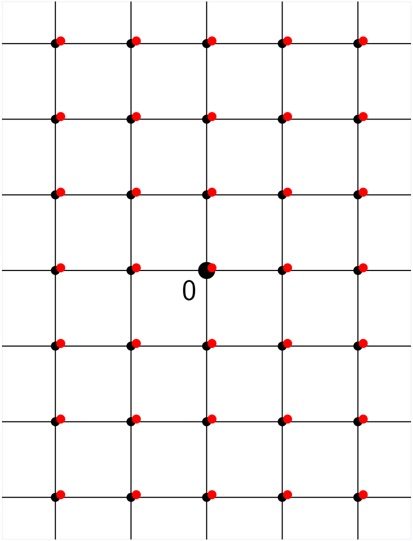
The dual of a lattice Λ is the set of all vectors $\mathbf{x} \in span(\Lambda)$ such that $\langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}$ for all $\mathbf{v} \in \Lambda$

Dual lattice: Examples



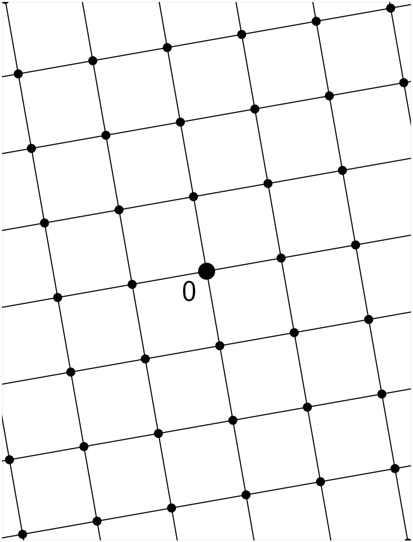
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



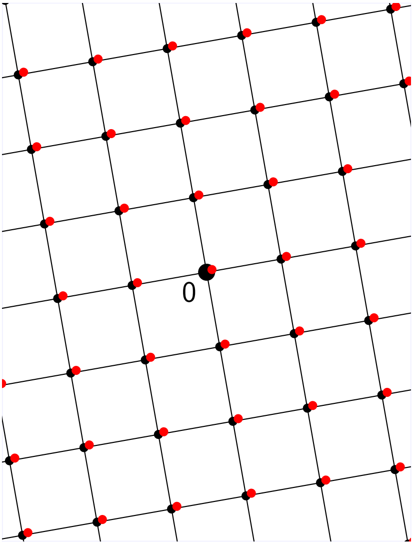
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



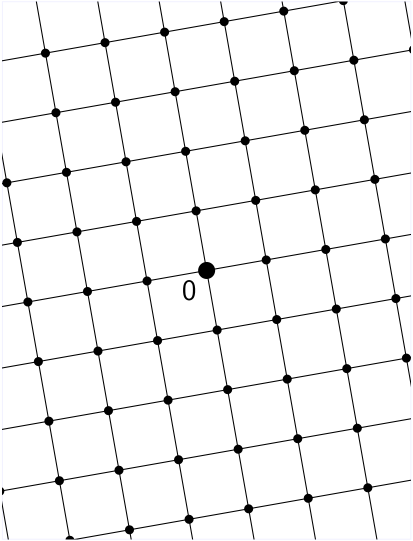
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



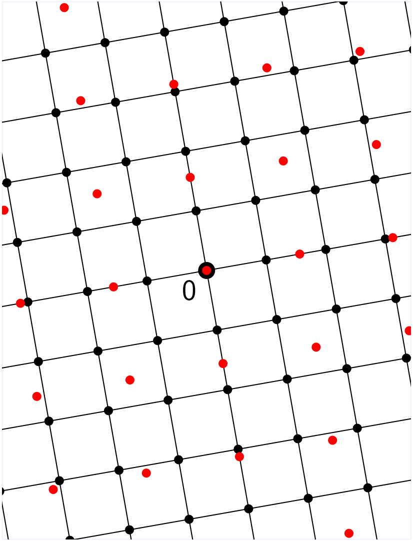
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



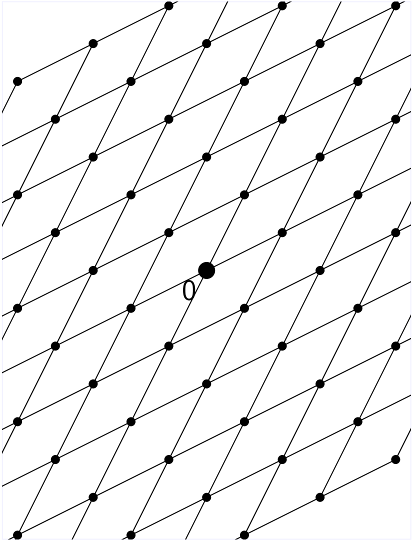
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



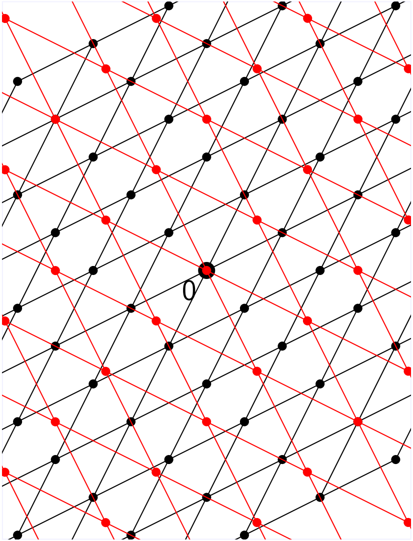
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



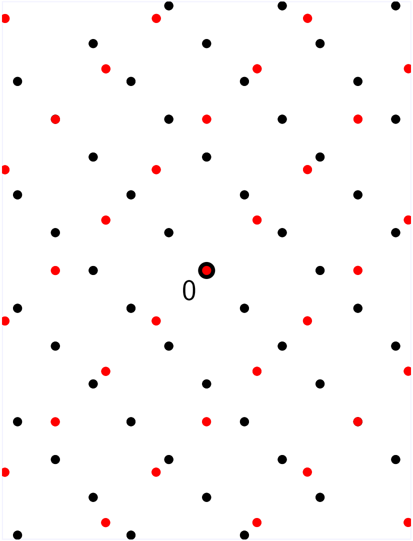
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



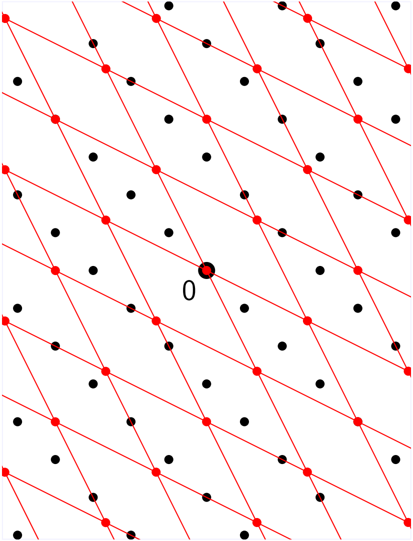
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



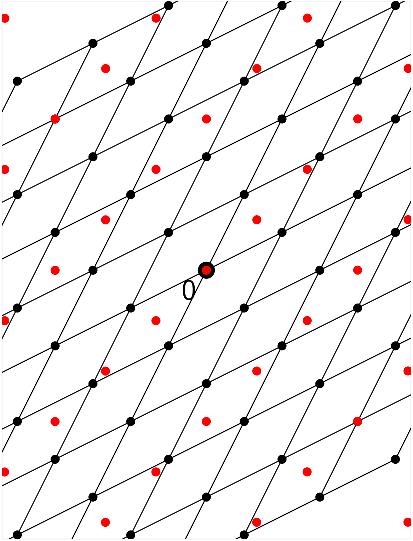
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



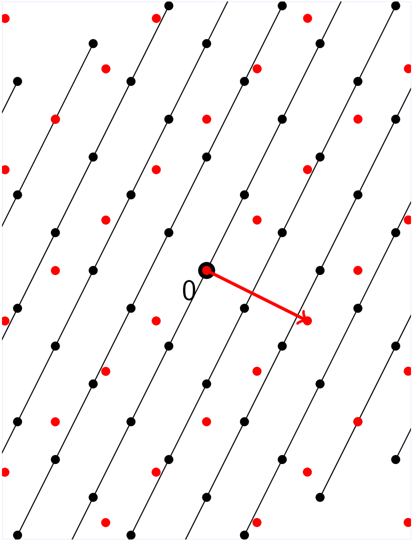
- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Dual lattice: Examples



- Integer lattice $(\mathbb{Z}^n)^* = \mathbb{Z}^n$
- Rotating $(\mathbf{R}\Lambda)^* = \mathbf{R}(\Lambda^*)$
- Scaling $(\frac{1}{q} \cdot \Lambda)^* = q \cdot \Lambda^*$
- Properties of dual:
  - $\Lambda_1 \subseteq \Lambda_2 \iff \Lambda_1^* \supseteq \Lambda_2^*$
  - $(\Lambda^*)^* = \Lambda$
- Operations on $\mathbf{x} \in \Lambda$ and $\mathbf{y} \in \Lambda^*$:
  - $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$
  - but $\mathbf{x} + \mathbf{y}$ has no geometric meaning

Lattice Layers

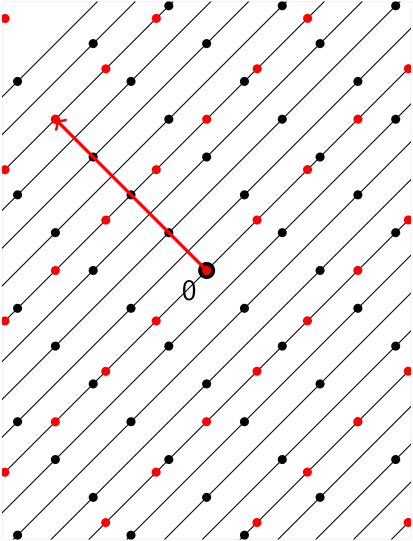


- Each dual vector $\mathbf{v} \in \mathcal{L}^*$, partitions the lattice $\mathcal{L}$ into layers orthogonal to $\mathbf{v}$

$$L_i = \{\mathbf{x} \in \mathcal{L} \mid \mathbf{x} \cdot \mathbf{v} = i\}$$

- Layers are at distance $1/\|\mathbf{v}\|$
- $\rho(\mathcal{L}) \geq \frac{1}{2\|\mathbf{v}\|}$
- If $\lambda_1(\mathcal{L}^*)$ is small, then $\rho(\mathcal{L})$ is large.

Lattice Layers

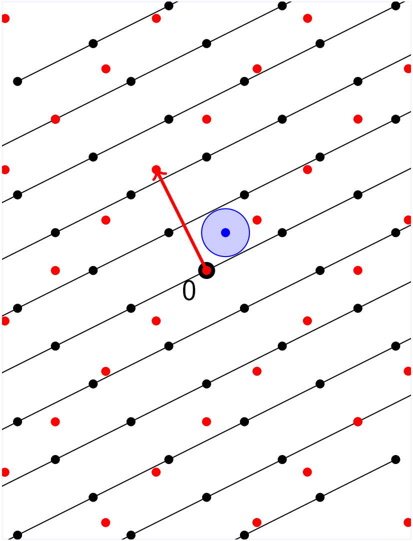


- Each dual vector $\mathbf{v} \in \mathcal{L}^*$, partitions the lattice $\mathcal{L}$ into layers orthogonal to $\mathbf{v}$

$$L_i = \{\mathbf{x} \in \mathcal{L} \mid \mathbf{x} \cdot \mathbf{v} = i\}$$

- Layers are at distance $1/\|\mathbf{v}\|$
- $\rho(\mathcal{L}) \geq \frac{1}{2\|\mathbf{v}\|}$
- If $\lambda_1(\mathcal{L}^*)$ is small, then $\rho(\mathcal{L})$ is large.

Lattice Layers

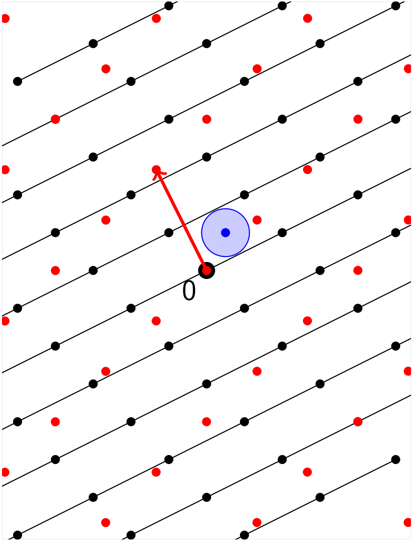


- Each dual vector $\mathbf{v} \in \mathcal{L}^*$, partitions the lattice $\mathcal{L}$ into layers orthogonal to $\mathbf{v}$

$$L_i = \{\mathbf{x} \in \mathcal{L} \mid \mathbf{x} \cdot \mathbf{v} = i\}$$

- Layers are at distance $1/\|\mathbf{v}\|$
- $\rho(\mathcal{L}) \geq \frac{1}{2\|\mathbf{v}\|}$
- If $\lambda_1(\mathcal{L}^*)$ is small, then $\rho(\mathcal{L})$ is large.

Lattice Layers



- Each dual vector $\mathbf{v} \in \mathcal{L}^*$, partitions the lattice $\mathcal{L}$ into layers orthogonal to $\mathbf{v}$

$$L_i = \{\mathbf{x} \in \mathcal{L} \mid \mathbf{x} \cdot \mathbf{v} = i\}$$

- Layers are at distance $1/\|\mathbf{v}\|$
- $\rho(\mathcal{L}) \geq \frac{1}{2\|\mathbf{v}\|}$
- If $\lambda_1(\mathcal{L}^*)$ is small, then $\rho(\mathcal{L})$ is large.

Transference Theorems

Theorem (Banaszczyk)

For any lattice $\mathcal{L}$

$$1 \leq 2\lambda_1(\mathcal{L}) \cdot \rho(\mathcal{L}^*) \leq n.$$

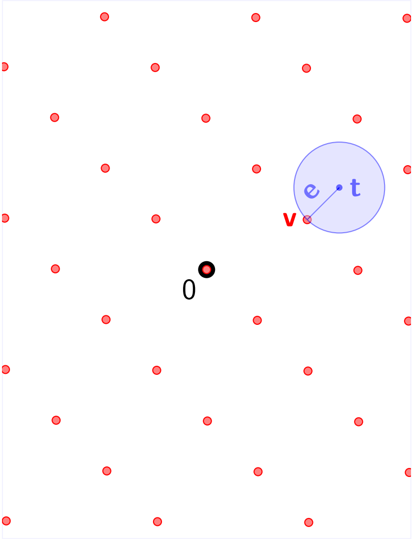
Theorem (Banaszczyk)

For every i,

$$1 \leq \lambda_i(\mathcal{L}) \cdot \lambda_{n-i+1}(\mathcal{L}^*) \leq n.$$

- Approximating $\lambda_1(\mathcal{L})$ within a factor n is in $NP \cap coNP$
- Same is true for $\lambda_i, \ldots, \lambda_n$ and ρ.

CVP and dual lattice



- Lattice Λ, target $\mathbf{t} = \mathbf{v} + \mathbf{e}$
- Dual lattice $\Lambda^* = \mathcal{L}(\mathbf{D})$.
- Syndrome of $\mathbf{t}$:

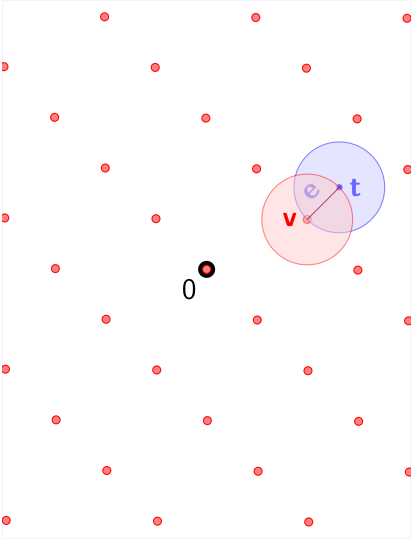
$$\begin{aligned} \mathbf{s} &= \langle \mathbf{D}, \mathbf{t} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{v} \rangle + \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1. \end{aligned}$$

- All vectors in a coset $\mathbf{t} + \mathcal{L}$ have the same syndrome.

Definition (Syndrome CVP)

Find shortest $\mathbf{e}$ such that $\langle \mathbf{D}, \mathbf{e} \rangle = \mathbf{s} \bmod 1$

CVP and dual lattice



- Lattice Λ, target $\mathbf{t} = \mathbf{v} + \mathbf{e}$
- Dual lattice $\Lambda^* = \mathcal{L}(\mathbf{D})$.
- Syndrome of $\mathbf{t}$:

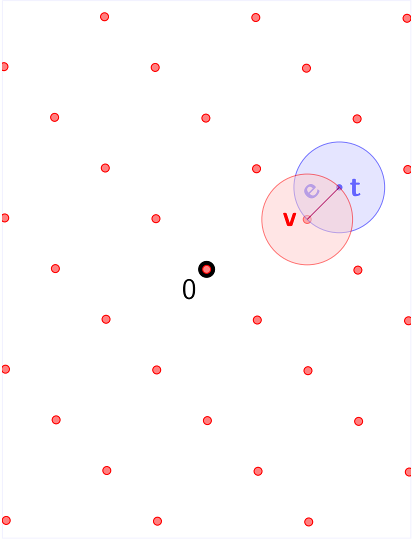
$$\begin{aligned} \mathbf{s} &= \langle \mathbf{D}, \mathbf{t} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{v} \rangle + \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1. \end{aligned}$$

- All vectors in a coset $\mathbf{t} + \mathcal{L}$ have the same syndrome.

Definition (Syndrome CVP)

Find shortest $\mathbf{e}$ such that $\langle \mathbf{D}, \mathbf{e} \rangle = \mathbf{s} \bmod 1$

CVP and dual lattice



- Lattice Λ, target $\mathbf{t} = \mathbf{v} + \mathbf{e}$
- Dual lattice $\Lambda^* = \mathcal{L}(\mathbf{D})$.
- Syndrome of $\mathbf{t}$:

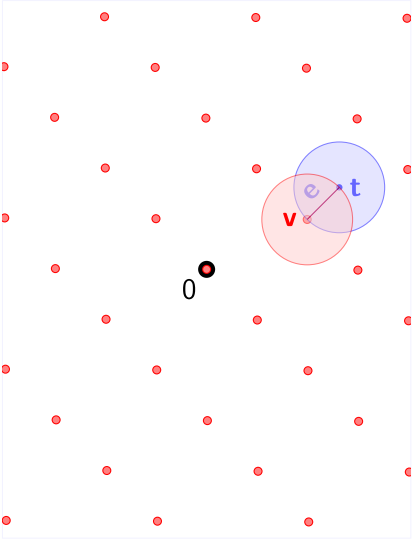
$$\begin{aligned} \mathbf{s} &= \langle \mathbf{D}, \mathbf{t} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{v} \rangle + \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1. \end{aligned}$$

- All vectors in a coset $\mathbf{t} + \mathcal{L}$ have the same syndrome.

Definition (Syndrome CVP)

Find shortest $\mathbf{e}$ such that $\langle \mathbf{D}, \mathbf{e} \rangle = \mathbf{s} \bmod 1$

CVP and dual lattice



- Lattice Λ, target $\mathbf{t} = \mathbf{v} + \mathbf{e}$
- Dual lattice $\Lambda^* = \mathcal{L}(\mathbf{D})$.
- Syndrome of $\mathbf{t}$:

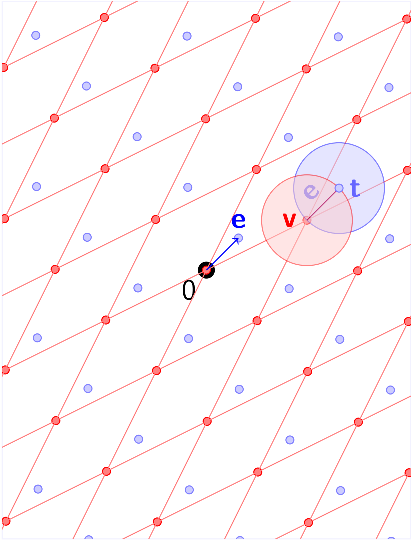
$$\begin{aligned} \mathbf{s} &= \langle \mathbf{D}, \mathbf{t} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{v} \rangle + \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1. \end{aligned}$$

- All vectors in a coset $\mathbf{t} + \mathcal{L}$ have the same syndrome.

Definition (Syndrome CVP)

Find shortest $\mathbf{e}$ such that $\langle \mathbf{D}, \mathbf{e} \rangle = \mathbf{s} \bmod 1$

CVP and dual lattice



- Lattice Λ, target $\mathbf{t} = \mathbf{v} + \mathbf{e}$
- Dual lattice $\Lambda^* = \mathcal{L}(\mathbf{D})$.
- Syndrome of $\mathbf{t}$:

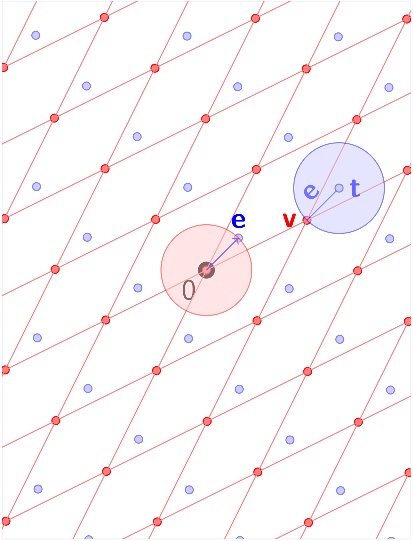
$$\begin{aligned}\mathbf{s} &= \langle \mathbf{D}, \mathbf{t} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{v} \rangle + \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1.\end{aligned}$$

- All vectors in a coset $\mathbf{t} + \mathcal{L}$ have the same syndrome.

Definition (Syndrome CVP)

Find shortest $\mathbf{e}$ such that $\langle \mathbf{D}, \mathbf{e} \rangle = \mathbf{s} \bmod 1$

CVP and dual lattice



- Lattice Λ, target $\mathbf{t} = \mathbf{v} + \mathbf{e}$
- Dual lattice $\Lambda^* = \mathcal{L}(\mathbf{D})$.
- Syndrome of $\mathbf{t}$:

$$\begin{aligned}\mathbf{s} &= \langle \mathbf{D}, \mathbf{t} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{v} \rangle + \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1 \\ &= \langle \mathbf{D}, \mathbf{e} \rangle \bmod 1.\end{aligned}$$

- All vectors in a coset $\mathbf{t} + \mathcal{L}$ have the same syndrome.

Definition (Syndrome CVP)

Find shortest $\mathbf{e}$ such that $\langle \mathbf{D}, \mathbf{e} \rangle = \mathbf{s} \bmod 1$

Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

Back to CVP One-way function

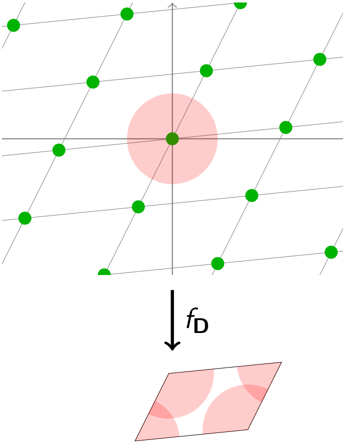
Candidate OWF

Key: a hard lattice $\mathcal{L}(\mathbf{D})^*$

Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$

Output: $f_{\mathbf{D}}(\mathbf{x}) = \mathbf{D}\mathbf{x} \bmod 1$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta \geq \mu$: $g_{\mathcal{L}}$ is surjective



Special Versions of CVP

Definition (Decisional CVP)

Given $(\mathcal{L}, \mathbf{t}, d)$, with $\mu(\mathbf{t}, \mathcal{L}) \leq d$, find a lattice point within distance d from $\mathbf{t}$.

- If d is arbitrary, then one can find the closest lattice vector by binary search on d.
- Bounded Distance Decoding, BDD: If $d < \lambda_1(\mathcal{L})/2$, then there is at most one solution. Solution is the closest lattice vector.
- Absolute Distance Decoding, ADD: If $d \geq \rho(\mathcal{L})$, then there is always at least one solution. Solution may not be closest lattice vector.

Special Versions of CVP

Definition (Decisional CVP)

Given $(\mathcal{L}, \mathbf{t}, d)$, with $\mu(\mathbf{t}, \mathcal{L}) \leq d$, find a lattice point within distance d from $\mathbf{t}$.

- If d is arbitrary, then one can find the closest lattice vector by binary search on d.
- Bounded Distance Decoding, BDD: If $d < \lambda_1(\mathcal{L})/2$, then there is at most one solution. Solution is the closest lattice vector.
- Absolute Distance Decoding, ADD: If $d \geq \rho(\mathcal{L})$, then there is always at least one solution. Solution may not be closest lattice vector.

Special Versions of CVP

Definition (Decisional CVP)

Given $(\mathcal{L}, \mathbf{t}, d)$, with $\mu(\mathbf{t}, \mathcal{L}) \leq d$, find a lattice point within distance d from $\mathbf{t}$.

- If d is arbitrary, then one can find the closest lattice vector by binary search on d.
- Bounded Distance Decoding, BDD: If $d < \lambda_1(\mathcal{L})/2$, then there is at most one solution. Solution is the closest lattice vector.
- Absolute Distance Decoding, ADD: If $d \geq \rho(\mathcal{L})$, then there is always at least one solution. Solution may not be closest lattice vector.

Special Versions of CVP

Definition (Decisional CVP)

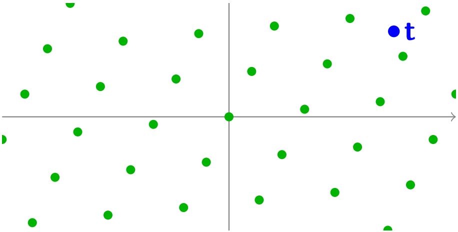
Given $(\mathcal{L}, \mathbf{t}, d)$, with $\mu(\mathbf{t}, \mathcal{L}) \leq d$, find a lattice point within distance d from $\mathbf{t}$.

- If d is arbitrary, then one can find the closest lattice vector by binary search on d.
- Bounded Distance Decoding, BDD: If $d < \lambda_1(\mathcal{L})/2$, then there is at most one solution. Solution is the closest lattice vector.
- Absolute Distance Decoding, ADD: If $d \geq \rho(\mathcal{L})$, then there is always at least one solution. Solution may not be closest lattice vector.

ADD reduces to SIVP

ADD input: $\mathcal{L}$ and arbitrary $\mathbf{t}$

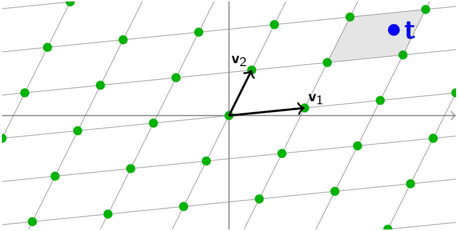
- Compute short vectors $\mathbf{V} = \mathrm{SIVP}(\mathcal{L})$
- Use $\mathbf{V}$ to find a lattice vector within distance $\sum_i \frac{1}{2}\|\mathbf{v}_i\| \leq (n/2)\lambda_n \leq n\rho$ from $\mathbf{t}$



ADD reduces to SIVP

ADD input: $\mathcal{L}$ and arbitrary $\mathbf{t}$

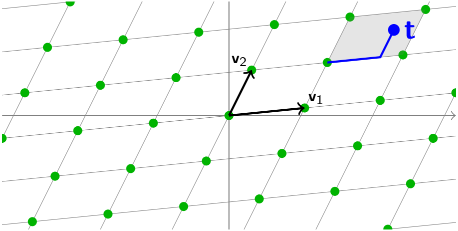
- Compute short vectors $\mathbf{V} = \mathrm{SIVP}(\mathcal{L})$
- Use $\mathbf{V}$ to find a lattice vector within distance $\sum_i \frac{1}{2}\|\mathbf{v}_i\| \leq (n/2)\lambda_n \leq n\rho$ from $\mathbf{t}$



ADD reduces to SIVP

ADD input: $\mathcal{L}$ and arbitrary $\mathbf{t}$

- Compute short vectors $\mathbf{V} = \mathrm{SIVP}(\mathcal{L})$
- Use $\mathbf{V}$ to find a lattice vector within distance $\sum_i \frac{1}{2}\|\mathbf{v}_i\| \leq (n/2)\lambda_n \leq n\rho$ from $\mathbf{t}$

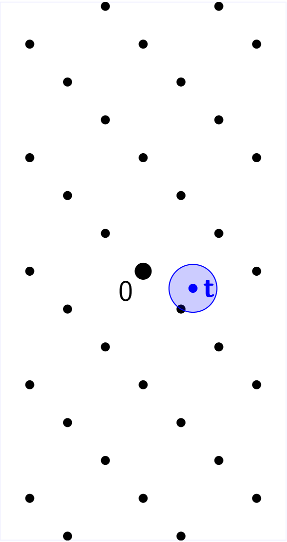


BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

- Compute $\mathbf{V} = \mathrm{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to $\mathbf{t}$
- Output $L_1 \cap L_2 \cap \cdots \cap L_n$
- Output is correct as long as

$$\mu(\mathbf{t}, \mathcal{L}) \leq \frac{\lambda_1}{2n} \leq \frac{1}{2\lambda_n^*} \leq \frac{1}{2\|\mathbf{v}_i\|}$$

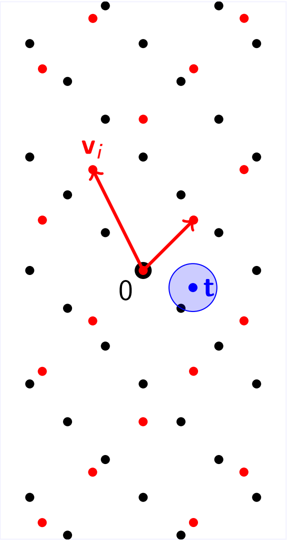


BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

- Compute $\mathbf{V} = \mathrm{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to $\mathbf{t}$
- Output $L_1 \cap L_2 \cap \cdots \cap L_n$
- Output is correct as long as

$$\mu(\mathbf{t}, \mathcal{L}) \leq \frac{\lambda_1}{2n} \leq \frac{1}{2\lambda_n^*} \leq \frac{1}{2\|\mathbf{v}_i\|}$$

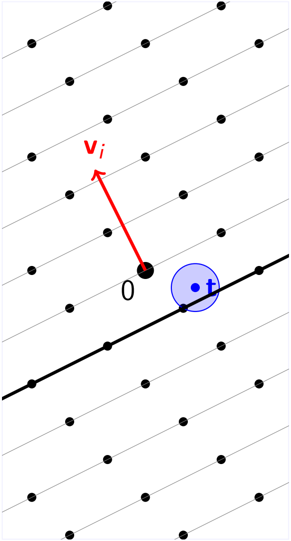


BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

- Compute $\mathbf{V} = \text{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to $\mathbf{t}$
- Output $L_1 \cap L_2 \cap \cdots \cap L_n$
- Output is correct as long as

$$\mu(\mathbf{t}, \mathcal{L}) \leq \frac{\lambda_1}{2n} \leq \frac{1}{2\lambda_n^*} \leq \frac{1}{2\|\mathbf{v}_i\|}$$

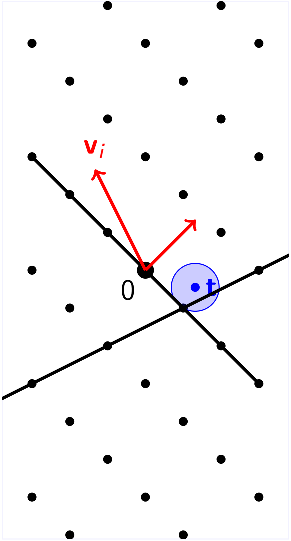


BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

- Compute $\mathbf{V} = \mathrm{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to $\mathbf{t}$
- Output $L_1 \cap L_2 \cap \cdots \cap L_n$
- Output is correct as long as

$$\mu(\mathbf{t}, \mathcal{L}) \leq \frac{\lambda_1}{2n} \leq \frac{1}{2\lambda_n^*} \leq \frac{1}{2\|\mathbf{v}_i\|}$$

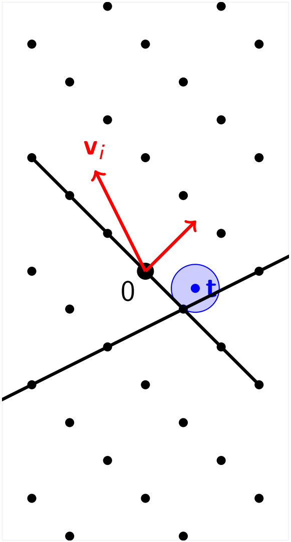


BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

- Compute $\mathbf{V} = \mathrm{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to $\mathbf{t}$
- Output $L_1 \cap L_2 \cap \cdots \cap L_n$
- Output is correct as long as

$$\mu(\mathbf{t}, \mathcal{L}) \leq \frac{\lambda_1}{2n} \leq \frac{1}{2\lambda_n^*} \leq \frac{1}{2\|\mathbf{v}_i\|}$$



Special Versions of SVP and SIVP

- GapSVP: compute (or approximate) the value λ_1 without necessarily finding a short vector
- GapSIVP: compute (or approximate) the value λ_n without necessarily finding short linearly independent vectors
- Transference Theorem $\lambda_1 \approx 1/\lambda_n^*$: GapSVP can be (approximately) solved by solving GapSIVP in the dual lattice, and vice versa

Problems

Exercise: Computing λ_1 (or λ_n) exactly is as hard as SVP (or SIVP)

Open Problem: Reduce approximate SVP (or SIVP) to approximate GapSVP (or GapSIVP)

Special Versions of SVP and SIVP

- GapSVP: compute (or approximate) the value λ_1 without necessarily finding a short vector
- GapSIVP: compute (or approximate) the value λ_n without necessarily finding short linearly independent vectors
- Transference Theorem $\lambda_1 \approx 1/\lambda_n^*$: GapSVP can be (approximately) solved by solving GapSIVP in the dual lattice, and vice versa

Problems

Exercise: Computing λ_1 (or λ_n) exactly is as hard as SVP (or SIVP)

Open Problem: Reduce approximate SVP (or SIVP) to approximate GapSVP (or GapSIVP)

Special Versions of SVP and SIVP

- GapSVP: compute (or approximate) the value λ_1 without necessarily finding a short vector
- GapSIVP: compute (or approximate) the value λ_n without necessarily finding short linearly independent vectors
- Transference Theorem $\lambda_1 \approx 1/\lambda_n^*$: GapSVP can be (approximately) solved by solving GapSIVP in the dual lattice, and vice versa

Problems

Exercise: Computing λ_1 (or λ_n) exactly is as hard as SVP (or SIVP)

Open Problem: Reduce approximate SVP (or SIVP) to approximate GapSVP (or GapSIVP)

Special Versions of SVP and SIVP

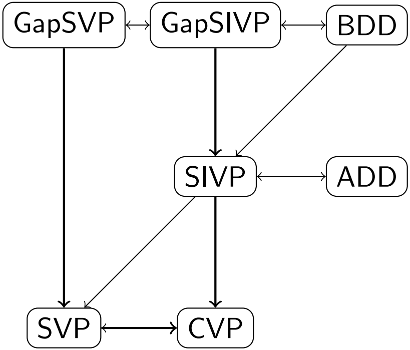
- GapSVP: compute (or approximate) the value λ_1 without necessarily finding a short vector
- GapSIVP: compute (or approximate) the value λ_n without necessarily finding short linearly independent vectors
- Transference Theorem $\lambda_1 \approx 1/\lambda_n^*$: GapSVP can be (approximately) solved by solving GapSIVP in the dual lattice, and vice versa

Problems

Exercise: Computing λ_1 (or λ_n) exactly is as hard as SVP (or SIVP)
Open Problem: Reduce approximate SVP (or SIVP) to approximate GapSVP (or GapSIVP)

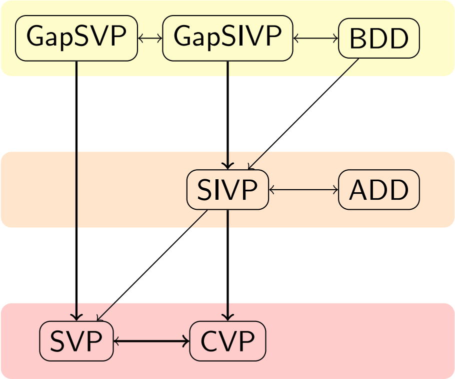
Relations among lattice problems

- SIVP $\approx$ ADD [MG'01]
- SVP $\leq$ CVP [GMSS'99]
- SIVP $\leq$ CVP [M'08]
- BDD $\lesssim$ SIVP
- CVP $\lesssim$ SVP [L'87]
- GapSVP $\approx$ GapSIVP [LLS'91,B'93]
- GapSVP $\lesssim$ BDD [LM'09]



Relations among lattice problems

- SIVP $\approx$ ADD [MG'01]
- SVP $\leq$ CVP [GMSS'99]
- SIVP $\leq$ CVP [M'08]
- BDD $\lesssim$ SIVP
- CVP $\lesssim$ SVP [L'87]
- GapSVP $\approx$ GapSIVP [LLS'91,B'93]
- GapSVP $\lesssim$ BDD [LM'09]



Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

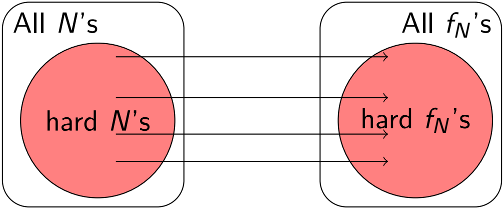
Provable security (from average case hardness)

Example 1: (Rabin) modular squaring

- $f_N(x) = x^2 \bmod N$, where $N = p \cdot q$
- Inverting f_N is at least as hard as factoring N

Theorem

f_N is cryptographically hard to invert, provided most $N = p \cdot q$ are hard to factor



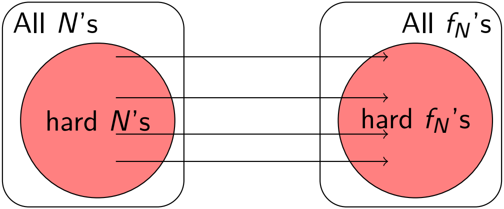
Provable security (from average case hardness)

Example 1: (Rabin) modular squaring

- $f_N(x) = x^2 \bmod N$, where $N = p \cdot q$
- Inverting f_N is at least as hard as factoring N

Theorem

f_N is cryptographically hard to invert, provided most $N = p \cdot q$ are hard to factor



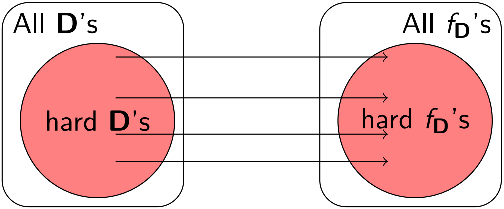
Provable security (from average case hardness)

Example 2: CVP function

- $f_{\mathbf{D}}(\mathbf{x}) = \mathbf{D}\mathbf{x} \bmod 1$
- Inverting $f_{\mathbf{D}}$ is as hard as ADD/BDD in $\mathcal{L}(\mathbf{D})^*$

Theorem

$f_{\mathbf{D}}$ is one-way provided ADD/BDD is hard for most $\mathcal{L}(\mathbf{D})^*$



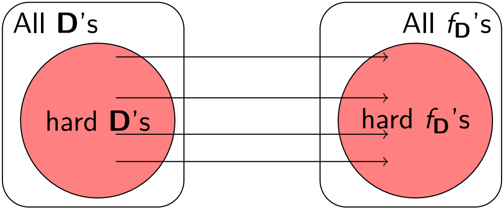
Provable security (from average case hardness)

Example 2: CVP function

- $f_{\mathbf{D}}(\mathbf{x}) = \mathbf{D}\mathbf{x} \bmod 1$
- Inverting $f_{\mathbf{D}}$ is as hard as ADD/BDD in $\mathcal{L}(\mathbf{D})^*$

Theorem

$f_{\mathbf{D}}$ is one-way provided ADD/BDD is hard for most $\mathcal{L}(\mathbf{D})^*$



Average-case Complexity

Average-case complexity depends on input distribution

Example (Factoring problem)

Given a number N, output $a, b > 1$ such that $N = ab$

Factoring can be easy on average

if N is uniformly random, then $N = 2 \cdot \frac{N}{2}$ with probability 50%!

- Factoring $N = pq$ is believed to be hard when p, q are randomly chosen primes
- How do we know $\mathcal{L}(\mathbf{D})^*$ is a hard distribution for ADD/BDD?

Average-case Complexity

Average-case complexity depends on input distribution

Example (Factoring problem)

Given a number N, output $a, b > 1$ such that $N = ab$

Factoring can be easy on average

if N is uniformly random, then $N = 2 \cdot \frac{N}{2}$ with probability 50%!

- Factoring $N = pq$ is believed to be hard when p, q are randomly chosen primes
- How do we know $\mathcal{L}(\mathbf{D})^*$ is a hard distribution for ADD/BDD?

Average-case Complexity

Average-case complexity depends on input distribution

Example (Factoring problem)

Given a number N, output $a, b > 1$ such that $N = ab$

Factoring can be easy on average

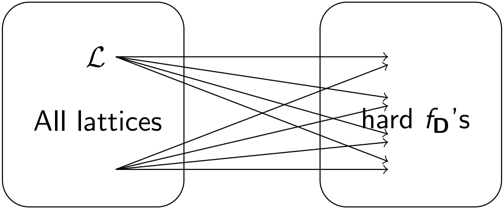
if N is uniformly random, then $N = 2 \cdot \frac{N}{2}$ with probability 50%!

- Factoring $N = pq$ is believed to be hard when p, q are randomly chosen primes
- How do we know $\mathcal{L}(\mathbf{D})^*$ is a hard distribution for ADD/BDD?

Provable security (from worst case hardness)

There is a probability distribution on **D** such that

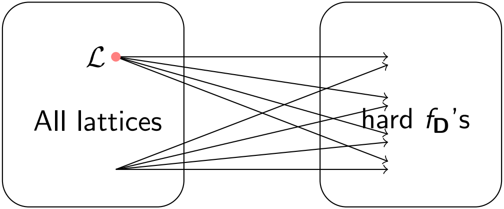
- Any fixed lattice $\mathcal{L}$ is mapped to a random **D**
- Breaking $f_{\mathbf{D}}$ allows to solve ADD/BDD $\mathcal{L}$.
- **D** is also very easy to sample



Provable security (from worst case hardness)

There is a probability distribution on **D** such that

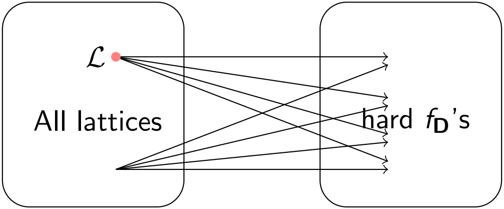
- Any fixed lattice $\mathcal{L}$ is mapped to a random **D**
- Breaking $f_{\mathbf{D}}$ allows to solve ADD/BDD $\mathcal{L}$.
- **D** is also very easy to sample



Provable security (from worst case hardness)

There is a probability distribution on **D** such that

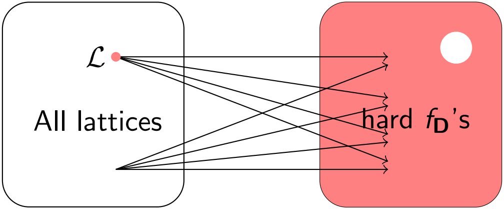
- Any fixed lattice $\mathcal{L}$ is mapped to a random **D**
- Breaking $f_{\mathbf{D}}$ allows to solve ADD/BDD $\mathcal{L}$.
- **D** is also very easy to sample



Provable security (from worst case hardness)

There is a probability distribution on **D** such that

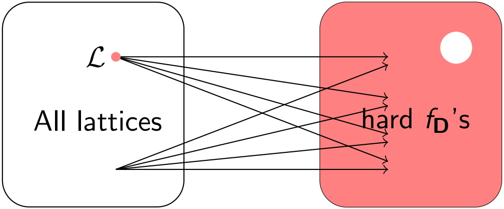
- Any fixed lattice $\mathcal{L}$ is mapped to a random **D**
- Breaking $f_{\mathbf{D}}$ allows to solve ADD/BDD $\mathcal{L}$.
- **D** is also very easy to sample



Provable security (from worst case hardness)

There is a probability distribution on $\mathbf{D}$ such that

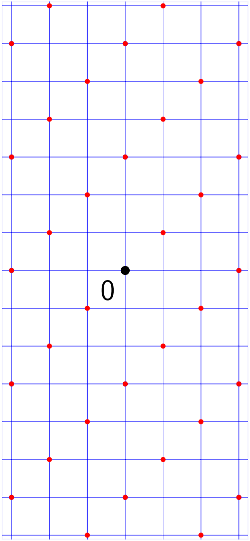
- Any fixed lattice $\mathcal{L}$ is mapped to a random $\mathbf{D}$
- Breaking $f_{\mathbf{D}}$ allows to solve ADD/BDD $\mathcal{L}$.
- $\mathbf{D}$ is also very easy to sample



Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

Random lattices in Cryptography

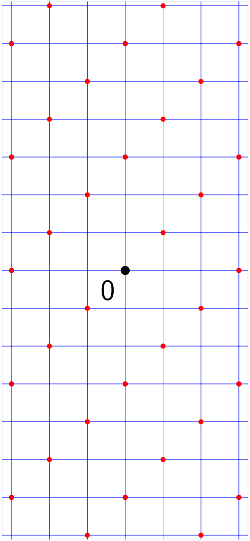


- Cryptography typically uses (random) lattices Λ such that
  - $\Lambda \subseteq \mathbb{Z}^d$ is an integer lattice
  - $q\mathbb{Z}^d \subseteq \Lambda$ is periodic modulo a small integer q.
- Cryptographic functions based on q-ary lattices involve only arithmetic modulo q.

Definition (q-ary lattice)

Λ is a q-ary lattice if $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$

Random lattices in Cryptography



- Cryptography typically uses (random) lattices Λ such that
  - $\Lambda \subseteq \mathbb{Z}^d$ is an integer lattice
  - $q\mathbb{Z}^d \subseteq \Lambda$ is periodic modulo a small integer q.
- Cryptographic functions based on q-ary lattices involve only arithmetic modulo q.

Definition (q-ary lattice)

Λ is a q-ary lattice if $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$

Examples of q-ary lattices

Examples (for any $\mathbf{A} \in \mathbb{Z}_q^{n \times d}$)

- $\Lambda_q(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{x} \bmod q \in \mathbf{A}^T \mathbb{Z}_q^n\} \subseteq \mathbb{Z}^d$
- $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\} \subseteq \mathbb{Z}^d$

Theorem

For any lattice Λ the following conditions are equivalent:

- $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$
- $\Lambda = \Lambda_q(\mathbf{A})$ *for some* $\mathbf{A}$
- $\Lambda = \Lambda_q^\perp(\mathbf{A})$ *for some* $\mathbf{A}$

For any fixed $\mathbf{A}$, the lattices $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are different

Examples of q-ary lattices

Examples (for any $\mathbf{A} \in \mathbb{Z}_q^{n\times d}$)

- $\Lambda_q(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{x} \bmod q \in \mathbf{A}^T\mathbb{Z}_q^n\} \subseteq \mathbb{Z}^d$
- $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\} \subseteq \mathbb{Z}^d$

Theorem

For any lattice Λ the following conditions are equivalent:

- $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$
- $\Lambda = \Lambda_q(\mathbf{A})$ *for some* $\mathbf{A}$
- $\Lambda = \Lambda_q^\perp(\mathbf{A})$ *for some* $\mathbf{A}$

For any fixed $\mathbf{A}$, the lattices $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are different

Examples of q-ary lattices

Examples (for any $\mathbf{A} \in \mathbb{Z}_q^{n \times d}$)

- $\Lambda_q(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{x} \bmod q \in \mathbf{A}^T \mathbb{Z}_q^n\} \subseteq \mathbb{Z}^d$
- $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\} \subseteq \mathbb{Z}^d$

Theorem

For any lattice Λ the following conditions are equivalent:

- $q\mathbb{Z}^d \subseteq \Lambda \subseteq \mathbb{Z}^d$
- $\Lambda = \Lambda_q(\mathbf{A})$ *for some* $\mathbf{A}$
- $\Lambda = \Lambda_q^\perp(\mathbf{A})$ *for some* $\mathbf{A}$

For any fixed $\mathbf{A}$, the lattices $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are different

Duality of q-ary lattices

- The q-ary lattices associated to $\mathbf{A}$ are dual (up to scaling)

$$\begin{aligned}\Lambda_q^\perp(\mathbf{A}) &= q \cdot \Lambda_q(\mathbf{A})^* \\ \Lambda_q(\mathbf{A}) &= q \cdot \Lambda_q^\perp(\mathbf{A})^*\end{aligned}$$

- In particular, $\det(\Lambda_q(\mathbf{A})) \cdot \det(\Lambda_q^\perp(\mathbf{A})) = q^n$
- $\det(\Lambda_q^\perp(\mathbf{A})) \le q^k$
- $\det(\Lambda_q(\mathbf{A})) \ge q^{n-k}$

Duality of q-ary lattices

- The q-ary lattices associated to $\mathbf{A}$ are dual (up to scaling)

$$\begin{aligned}\Lambda_q^{\perp}(\mathbf{A}) &= q \cdot \Lambda_q(\mathbf{A})^* \\ \Lambda_q(\mathbf{A}) &= q \cdot \Lambda_q^{\perp}(\mathbf{A})^*\end{aligned}$$

- In particular, $\det(\Lambda_q(\mathbf{A})) \cdot \det(\Lambda_q^{\perp}(\mathbf{A})) = q^n$
- $\det(\Lambda_q^{\perp}(\mathbf{A})) \leq q^k$
- $\det(\Lambda_q(\mathbf{A})) \geq q^{n-k}$

Duality of q-ary lattices

- The q-ary lattices associated to $\mathbf{A}$ are dual (up to scaling)

$$\begin{aligned}\Lambda_q^{\perp}(\mathbf{A}) &= q \cdot \Lambda_q(\mathbf{A})^* \\ \Lambda_q(\mathbf{A}) &= q \cdot \Lambda_q^{\perp}(\mathbf{A})^*\end{aligned}$$

- In particular, $\det(\Lambda_q(\mathbf{A})) \cdot \det(\Lambda_q^{\perp}(\mathbf{A})) = q^n$
- $\det(\Lambda_q^{\perp}(\mathbf{A})) \leq q^k$
- $\det(\Lambda_q(\mathbf{A})) \geq q^{n-k}$

Non-degenerate Matrices

Definition

$$\mathcal{M}_{k,n} = \{\mathbf{A} \in \mathbb{Z}_q^{k\times n} \mid \mathbf{A}\mathbb{Z}_q^n = \mathbb{Z}_q^k\}$$

- $\Pr\{\mathbf{A} \in \mathcal{M}_{k,n}\} \geq 1 - \frac{1}{q^{n-k}}$
- $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ are the same distribution
- $\det(\Lambda_q^\perp(\mathcal{M}_{k,n})) = \det(\Lambda_q(\mathcal{M}_{n-k,n})) = q^k$
- Minkowki's bound $\lambda_1 \leq \sqrt{n}q^{k/n}$

Theorem

Almost every lattice in $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ satisfies

$$\lambda_1, \ldots, \lambda_n, \rho = \Theta(\sqrt{n}q^{k,n})$$

Non-degenerate Matrices

Definition

$$\mathcal{M}_{k,n} = \{\mathbf{A} \in \mathbb{Z}_q^{k\times n} \mid \mathbf{A}\mathbb{Z}_q^n = \mathbb{Z}_q^k\}$$

- $\Pr\{\mathbf{A} \in \mathcal{M}_{k,n}\} \geq 1 - \frac{1}{q^{n-k}}$
- $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ are the same distribution
- $\det(\Lambda_q^\perp(\mathcal{M}_{k,n})) = \det(\Lambda_q(\mathcal{M}_{n-k,n})) = q^k$
- Minkowki's bound $\lambda_1 \leq \sqrt{n} q^{k/n}$

Theorem

Almost every lattice in $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ satisfies

$$\lambda_1, \ldots, \lambda_n, \rho = \Theta(\sqrt{n} q^{k,n})$$

Non-degenerate Matrices

Definition

$$\mathcal{M}_{k,n} = \{\mathbf{A} \in \mathbb{Z}_q^{k\times n} \mid \mathbf{A}\mathbb{Z}_q^n = \mathbb{Z}_q^k\}$$

- $\Pr\{\mathbf{A} \in \mathcal{M}_{k,n}\} \geq 1 - \frac{1}{q^{n-k}}$
- $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ are the same distribution
- $\det(\Lambda_q^\perp(\mathcal{M}_{k,n})) = \det(\Lambda_q(\mathcal{M}_{n-k,n})) = q^k$
- Minkowki's bound $\lambda_1 \leq \sqrt{n}q^{k/n}$

Theorem

Almost every lattice in $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ satisfies

$$\lambda_1,\ldots,\lambda_n,\rho = \Theta(\sqrt{n}q^{k,n})$$

Non-degenerate Matrices

Definition

$$\mathcal{M}_{k,n} = \{\mathbf{A} \in \mathbb{Z}_q^{k\times n} \mid \mathbf{A}\mathbb{Z}_q^n = \mathbb{Z}_q^k\}$$

- $\Pr\{\mathbf{A} \in \mathcal{M}_{k,n}\} \geq 1 - \frac{1}{q^{n-k}}$
- $\Lambda_q^{\perp}(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ are the same distribution
- $\det(\Lambda_q^{\perp}(\mathcal{M}_{k,n})) = \det(\Lambda_q(\mathcal{M}_{n-k,n})) = q^k$
- Minkowski's bound $\lambda_1 \leq \sqrt{n}q^{k/n}$

Theorem

Almost every lattice in $\Lambda_q^{\perp}(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ satisfies

$$\lambda_1, \ldots, \lambda_n, \rho = \Theta(\sqrt{n}q^{k,n})$$

Non-degenerate Matrices

Definition

$$\mathcal{M}_{k,n} = \{\mathbf{A} \in \mathbb{Z}_q^{k \times n} \mid \mathbf{A}\mathbb{Z}_q^n = \mathbb{Z}_q^k\}$$

- $\Pr\{\mathbf{A} \in \mathcal{M}_{k,n}\} \geq 1 - \frac{1}{q^{n-k}}$
- $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ are the same distribution
- $\det(\Lambda_q^\perp(\mathcal{M}_{k,n})) = \det(\Lambda_q(\mathcal{M}_{n-k,n})) = q^k$
- Minkowki's bound $\lambda_1 \leq \sqrt{n} q^{k/n}$

Theorem

Almost every lattice in $\Lambda_q^\perp(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ satisfies

$$\lambda_1, \ldots, \lambda_n, \rho = \Theta(\sqrt{n} q^{k,n})$$

Non-degenerate Matrices

Definition

$$\mathcal{M}_{k,n} = \{\mathbf{A} \in \mathbb{Z}_q^{k\times n} \mid \mathbf{A}\mathbb{Z}_q^n = \mathbb{Z}_q^k\}$$

- $\Pr\{\mathbf{A} \in \mathcal{M}_{k,n}\} \geq 1 - \frac{1}{q^{n-k}}$
- $\Lambda_q^{\perp}(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ are the same distribution
- $\det(\Lambda_q^{\perp}(\mathcal{M}_{k,n})) = \det(\Lambda_q(\mathcal{M}_{n-k,n})) = q^k$
- Minkowki's bound $\lambda_1 \leq \sqrt{n}q^{k/n}$

Theorem

Almost every lattice in $\Lambda_q^{\perp}(\mathcal{M}_{k,n}) \equiv \Lambda_q(\mathcal{M}_{n-k,n})$ satisfies

$$\lambda_1, \ldots, \lambda_n, \rho = \Theta(\sqrt{n}q^{k,n})$$

Are q-ary lattices hard?

Question

Are lattice problems on random q-ary lattices hard on average?

- GapSVP and GapSIVP are easy!
- Why? Just output Minkowki's bound $\sqrt{n}q^{k/n}$!
- What about BDD? (Remember BDD $\leq$ GapSVP.)
- BDD may still be hard! Reduction from BDD to GapSVP requires a wost-case GapSVP oracle.
- Are ADD, SIVP, SVP, CVP hard?

Are q-ary lattices hard?

Question

Are lattice problems on random q-ary lattices hard on average?

- GapSVP and GapSIVP are easy!
- Why? Just output Minkowki's bound $\sqrt{n}q^{k/n}$!
- What about BDD? (Remember BDD $\leq$ GapSVP.)
- BDD may still be hard! Reduction from BDD to GapSVP requires a wost-case GapSVP oracle.
- Are ADD, SIVP, SVP, CVP hard?

Are q-ary lattices hard?

Question

Are lattice problems on random q-ary lattices hard on average?

- GapSVP and GapSIVP are easy!
- Why? Just output Minkowki's bound $\sqrt{n}q^{k/n}$!
- What about BDD? (Remember BDD $\leq$ GapSVP.)
- BDD may still be hard! Reduction from BDD to GapSVP requires a wost-case GapSVP oracle.
- Are ADD, SIVP, SVP, CVP hard?

Are q-ary lattices hard?

Question

Are lattice problems on random q-ary lattices hard on average?

- GapSVP and GapSIVP are easy!
- Why? Just output Minkowki's bound $\sqrt{n}q^{k/n}$!
- What about BDD? (Remember BDD $\leq$ GapSVP.)
- BDD may still be hard! Reduction from BDD to GapSVP requires a wost-case GapSVP oracle.
- Are ADD, SIVP, SVP, CVP hard?

Are q-ary lattices hard?

Question

Are lattice problems on random q-ary lattices hard on average?

- GapSVP and GapSIVP are easy!
- Why? Just output Minkowki's bound $\sqrt{n}q^{k/n}$!
- What about BDD? (Remember BDD $\leq$ GapSVP.)
- BDD may still be hard! Reduction from BDD to GapSVP requires a wost-case GapSVP oracle.
- Are ADD, SIVP, SVP, CVP hard?

Are q-ary lattices hard?

Question

Are lattice problems on random q-ary lattices hard on average?

- GapSVP and GapSIVP are easy!
- Why? Just output Minkowki's bound $\sqrt{n}q^{k/n}$!
- What about BDD? (Remember BDD $\leq$ GapSVP.)
- BDD may still be hard! Reduction from BDD to GapSVP requires a wost-case GapSVP oracle.
- Are ADD, SIVP, SVP, CVP hard?

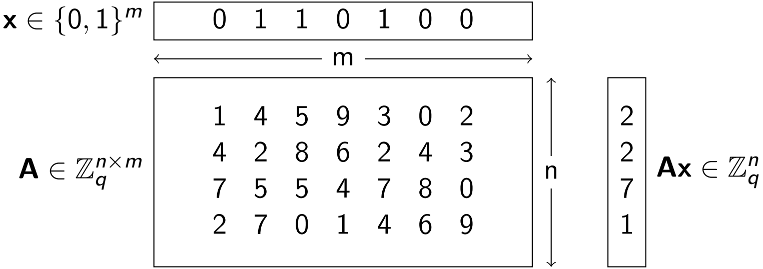
Ajtai's function

Definition (Ajtai's function)

Keyed function family

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$$

where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{x} \in \{0,1\}^m$.



$$\mathbf{x} \in \{0,1\}^m \quad \boxed{\begin{matrix}0 & 1 & 1 & 0 & 1 & 0 & 0\end{matrix}}$$

$$\xleftrightarrow{\quad m \quad}$$

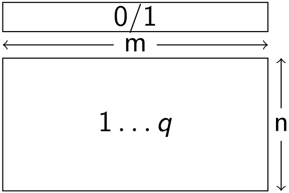
$$\mathbf{A} \in \mathbb{Z}_q^{n\times m} \quad \boxed{\begin{matrix}1 & 4 & 5 & 9 & 3 & 0 & 2\\ 4 & 2 & 8 & 6 & 2 & 4 & 3\\ 7 & 5 & 5 & 4 & 7 & 8 & 0\\ 2 & 7 & 0 & 1 & 4 & 6 & 9\end{matrix}} \quad \updownarrow n \quad \boxed{\begin{matrix}2\\ 2\\ 7\\ 1\end{matrix}} \quad \mathbf{Ax} \in \mathbb{Z}_q^n$$

Ajtai's function and q-ary lattices

- $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$, where $\mathbf{x}$ is short
- The output of $f_{\mathbf{A}}(\mathbf{x})$ is the syndrome of $\mathbf{x}$
- Inverting $f_{\mathbf{A}}(\mathbf{x})$ is the same as CVP in its syndrome decoding formulation with lattice $\Lambda_q^{\perp}(\mathbf{A})$ and target $\mathbf{t} \in \mathbf{x} + \Lambda_q^{\perp}(\mathbf{A})$
- The q-ary lattice $\Lambda_q^{\perp}(\mathbf{A})$ is the kernel of $f_{\mathbf{A}}$
- Finding collisions $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{y})$ is equivalent to finding short vectors $\mathbf{x} - \mathbf{y} \in \Lambda_q^{\perp}(\mathbf{A})$

Parameters

- Parameters:
  - n: main security parameter
  - $q = n^2 = n^{O(1)}$ small modulus
  - $m = 2n \log_2 q = O(n \log n)$
  - e.g., $n = 256$, $q = 2^{16}$, $m = 8192$
- $f_{\mathbf{A}}$ is a compression function
  - It maps m bits to $n \log_2 q < m$ bits (e.g., $8192 \to 4096$)
  - There exist collisions $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{y})$


$0/1$

m

$1 \dots q$

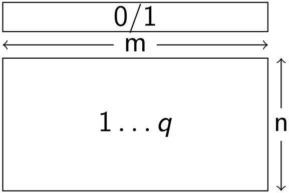
n

Question

Is $f_{\mathbf{A}}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is chosen at random?

Parameters

- Parameters:
  - n: main security parameter
  - $q = n^2 = n^{O(1)}$ small modulus
  - $m = 2n \log_2 q = O(n \log n)$
  - e.g., $n = 256$, $q = 2^{16}$, $m = 8192$
- $f_{\mathbf{A}}$ is a compression function
  - It maps m bits to $n \log_2 q < m$ bits (e.g., $8192 \rightarrow 4096$)
  - There exist collisions $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{y})$

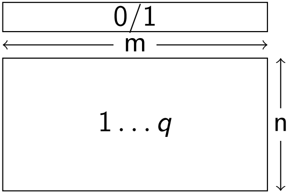


Question

Is $f_{\mathbf{A}}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is chosen at random?

Parameters

- Parameters:
  - n: main security parameter
  - $q = n^2 = n^{O(1)}$ small modulus
  - $m = 2n\log_2 q = O(n\log n)$
  - e.g., $n = 256$, $q = 2^{16}$, $m = 8192$
- $f_{\mathbf{A}}$ is a compression function
  - It maps m bits to $n\log_2 q < m$ bits (e.g., $8192 \to 4096$)
  - There exist collisions $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{y})$

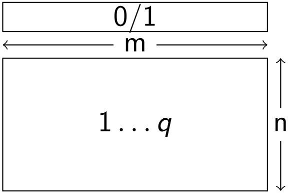


Question

Is $f_{\mathbf{A}}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is chosen at random?

Parameters

- Parameters:
  - n: main security parameter
  - $q = n^2 = n^{O(1)}$ small modulus
  - $m = 2n \log_2 q = O(n \log n)$
  - e.g., $n = 256$, $q = 2^{16}$, $m = 8192$
- $f_{\mathbf{A}}$ is a compression function
  - It maps m bits to $n \log_2 q < m$ bits (e.g., $8192 \rightarrow 4096$)
  - There exist collisions $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{y})$

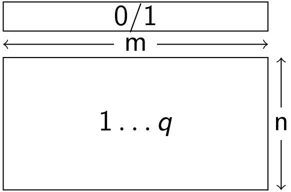


Question

Is $f_{\mathbf{A}}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is chosen at random?

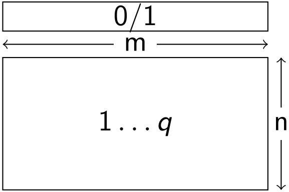
Efficiency issues

- $q = n^{O(1)}$, $m = 2n \log_2 q$
- Let's lower $n = 64$, $q = 2^8$, $m = 1024$
- $f_{\mathbf{A}}$ maps 1024 bits to 512.
- Key size: $nm \log q = O(n^2 \log^2 n) = 2^{19} = 64KB$
- Runtime: $nm = O(n^2 \log n) = 2^{16}$ arithmetic operations
- Still inefficient because of quadratic dependency in n



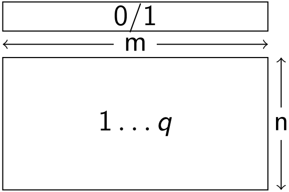
Efficiency issues

- $q = n^{O(1)}$, $m = 2n \log_2 q$
- Let's lower $n = 64$, $q = 2^8$, $m = 1024$
- $f_{\mathbf{A}}$ maps 1024 bits to 512.
- Key size: $nm \log q = O(n^2 \log^2 n) = 2^{19} = 64KB$
- Runtime: $nm = O(n^2 \log n) = 2^{16}$ arithmetic operations
- Still inefficient because of quadratic dependency in n



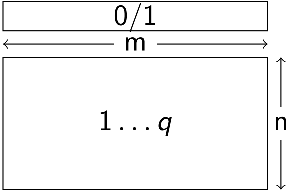
Efficiency issues

- $q = n^{O(1)}$, $m = 2n \log_2 q$
- Let's lower $n = 64$, $q = 2^8$, $m = 1024$
- $f_{\mathbf{A}}$ maps 1024 bits to 512.
- Key size: $nm \log q = O(n^2 \log^2 n) = 2^{19} = 64KB$
- Runtime: $nm = O(n^2 \log n) = 2^{16}$ arithmetic operations
- Still inefficient because of quadratic dependency in n



Efficiency issues

- $q = n^{O(1)}$, $m = 2n \log_2 q$
- Let's lower $n = 64$, $q = 2^8$, $m = 1024$
- $f_{\mathbf{A}}$ maps 1024 bits to 512.
- Key size: $nm \log q = O(n^2 \log^2 n) = 2^{19} = 64KB$
- Runtime: $nm = O(n^2 \log n) = 2^{16}$ arithmetic operations
- Still inefficient because of quadratic dependency in n



Efficient lattice based hashing

Idea

Use structured matrix

$$\mathbf{A} = [\mathbf{A}^{(1)} \mid \ldots \mid \mathbf{A}^{(m/n)}]$$

where $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{n \times n}$ is circulant

$$\mathbf{A}^{(i)} = \begin{bmatrix} a_1^{(i)} & a_n^{(i)} & \cdots & a_2^{(i)} \\ a_2^{(i)} & a_1^{(i)} & \cdots & a_3^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ a_n^{(i)} & a_{n-1}^{(i)} & \cdots & a_1^{(i)} \end{bmatrix}$$

- Proposed by [M02], where it is proved that $f_{\mathbf{A}}$ is one-way under plausible complexity assumptions
- Similar idea first used by NTRU public key cryptosystem (1998), but with no proof of security
- Wishful thinking: finding short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ is hard, and therefore $f_{\mathbf{A}}$ is collision resistant

Efficient lattice based hashing

Idea

Use structured matrix

$$\mathbf{A} = [\mathbf{A}^{(1)} \mid \dots \mid \mathbf{A}^{(m/n)}]$$

where $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{n\times n}$ is circulant

$$\mathbf{A}^{(i)} = \begin{bmatrix} a_1^{(i)} & a_n^{(i)} & \cdots & a_2^{(i)} \\ a_2^{(i)} & a_1^{(i)} & \cdots & a_3^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ a_n^{(i)} & a_{n-1}^{(i)} & \cdots & a_1^{(i)} \end{bmatrix}$$

- Proposed by [M02], where it is proved that $f_{\mathbf{A}}$ is one-way under plausible complexity assumptions
- Similar idea first used by NTRU public key cryptosystem (1998), but with no proof of security
- Wishful thinking: finding short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ is hard, and therefore $f_{\mathbf{A}}$ is collision resistant

Efficient lattice based hashing

Idea

Use structured matrix

$$\mathbf{A} = [\mathbf{A}^{(1)} \mid \ldots \mid \mathbf{A}^{(m/n)}]$$

where $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{n \times n}$ is circulant

$$\mathbf{A}^{(i)} = \begin{bmatrix} a_1^{(i)} & a_n^{(i)} & \cdots & a_2^{(i)} \\ a_2^{(i)} & a_1^{(i)} & \cdots & a_3^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ a_n^{(i)} & a_{n-1}^{(i)} & \cdots & a_1^{(i)} \end{bmatrix}$$

- Proposed by [M02], where it is proved that $f_{\mathbf{A}}$ is one-way under plausible complexity assumptions
- Similar idea first used by NTRU public key cryptosystem (1998), but with no proof of security
- Wishful thinking: finding short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ is hard, and therefore $f_{\mathbf{A}}$ is collision resistant

Efficient lattice based hashing

Idea

Use structured matrix

$$\mathbf{A} = [\mathbf{A}^{(1)} \mid \ldots \mid \mathbf{A}^{(m/n)}]$$

where $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{n\times n}$ is circulant

$$\mathbf{A}^{(i)} = \begin{bmatrix} a_1^{(i)} & a_n^{(i)} & \cdots & a_2^{(i)} \\ a_2^{(i)} & a_1^{(i)} & \cdots & a_3^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ a_n^{(i)} & a_{n-1}^{(i)} & \cdots & a_1^{(i)} \end{bmatrix}$$

- Proposed by [M02], where it is proved that $f_{\mathbf{A}}$ is one-way under plausible complexity assumptions
- Similar idea first used by NTRU public key cryptosystem (1998), but with no proof of security
- Wishful thinking: finding short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ is hard, and therefore $f_{\mathbf{A}}$ is collision resistant

Can you find a collision?

1	4	3	8	6	4	9	0	2	6	4	5	3	2	7	1
8	1	4	3	0	6	4	9	5	2	6	4	1	3	2	7
3	8	1	4	9	0	6	4	4	5	2	6	7	1	3	2
4	3	8	1	4	9	0	6	6	4	5	2	2	7	1	3

Can you find a collision?

1	0	0	-1	-1	1	1	0	0	0	1	1	1	0	-1	0	
1	4	3	8	6	4	9	0	2	6	4	5	3	2	7	1	5
8	1	4	3	0	6	4	9	5	2	6	4	1	3	2	7	4
3	8	1	4	9	0	6	4	4	5	2	6	7	1	3	2	8
4	3	8	1	4	9	0	6	6	4	5	2	2	7	1	3	6

Can you find a collision?

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	
1	4	3	8	6	4	9	0	2	6	4	5	3	2	7	1	0
8	1	4	3	0	6	4	9	5	2	6	4	1	3	2	7	0
3	8	1	4	9	0	6	4	4	5	2	6	7	1	3	2	0
4	3	8	1	4	9	0	6	6	4	5	2	2	7	1	3	0

Can you find a collision?

$$
\left|\begin{array}{cccc|cccc|cccc|cccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
1 & 4 & 3 & 8 & 6 & 4 & 9 & 0 & 2 & 6 & 4 & 5 & 3 & 2 & 7 & 1 \\
8 & 1 & 4 & 3 & 0 & 6 & 4 & 9 & 5 & 2 & 6 & 4 & 1 & 3 & 2 & 7 \\
3 & 8 & 1 & 4 & 9 & 0 & 6 & 4 & 4 & 5 & 2 & 6 & 7 & 1 & 3 & 2 \\
4 & 3 & 8 & 1 & 4 & 9 & 0 & 6 & 6 & 4 & 5 & 2 & 2 & 7 & 1 & 3 \\
\hline
\end{array}\right|
\qquad
\left|\begin{array}{c}
\ \\
\hline
\ \\
\ \\
\ \\
\ \\
\hline
\end{array}\right|
$$

$$
+1\times\begin{array}{|c|}\hline 6\\ 6\\ 6\\ 6\\ \hline\end{array}
\qquad
-1\times\begin{array}{|c|}\hline 9\\ 9\\ 9\\ 9\\ \hline\end{array}
\qquad
+0\times\begin{array}{|c|}\hline 7\\ 7\\ 7\\ 7\\ \hline\end{array}
\qquad
+1\times\begin{array}{|c|}\hline 3\\ 3\\ 3\\ 3\\ \hline\end{array}
$$

Can you find a collision?

$$
\begin{array}{|cccc|cccc|cccc|cccc|}
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
1 & 4 & 3 & 8 & 6 & 4 & 9 & 0 & 2 & 6 & 4 & 5 & 3 & 2 & 7 & 1 \\
8 & 1 & 4 & 3 & 0 & 6 & 4 & 9 & 5 & 2 & 6 & 4 & 1 & 3 & 2 & 7 \\
3 & 8 & 1 & 4 & 9 & 0 & 6 & 4 & 4 & 5 & 2 & 6 & 7 & 1 & 3 & 2 \\
4 & 3 & 8 & 1 & 4 & 9 & 0 & 6 & 6 & 4 & 5 & 2 & 2 & 7 & 1 & 3 \\
\hline
\end{array}
\qquad
\begin{array}{|c|}
\\
\hline
0 \\
0 \\
0 \\
0 \\
\hline
\end{array}
$$

$$
+1 \times \begin{array}{|c|} \hline 6 \\ 6 \\ 6 \\ 6 \\ \hline \end{array}
\qquad
-1 \times \begin{array}{|c|} \hline 9 \\ 9 \\ 9 \\ 9 \\ \hline \end{array}
\qquad
+0 \times \begin{array}{|c|} \hline 7 \\ 7 \\ 7 \\ 7 \\ \hline \end{array}
\qquad
+1 \times \begin{array}{|c|} \hline 3 \\ 3 \\ 3 \\ 3 \\ \hline \end{array}
$$

Remarks about proofs of security

- This function is essentially the compression function of hash function LASH, modeled after NTRU
- You can still "prove" security based on average case assumption: Breaking the above hash function is as hard as finding short vectors in a random lattice $\Lambda([\mathbf{A}^{(1)}|\ldots|\mathbf{A}^{(m/n)}])$
- ...but we know the function is broken: The underlying random lattice distribution is weak!
- Conclusion: Assuming that a problem is hard on average-case is a really tricky business!

Remarks about proofs of security

- This function is essentially the compression function of hash function LASH, modeled after NTRU
- You can still "prove" security based on average case assumption: Breaking the above hash function is as hard as finding short vectors in a random lattice $\Lambda([\mathbf{A}^{(1)}|\ldots|\mathbf{A}^{(m/n)}])$
- . . . but we know the function is broken: The underlying random lattice distribution is weak!
- Conclusion: Assuming that a problem is hard on average-case is a really tricky business!

Remarks about proofs of security

- This function is essentially the compression function of hash function LASH, modeled after NTRU
- You can still "prove" security based on average case assumption: Breaking the above hash function is as hard as finding short vectors in a random lattice $\Lambda([\mathbf{A}^{(1)}|\ldots|\mathbf{A}^{(m/n)}])$
- ...but we know the function is broken: The underlying random lattice distribution is weak!
- Conclusion: Assuming that a problem is hard on average-case is a really tricky business!

Remarks about proofs of security

- This function is essentially the compression function of hash function LASH, modeled after NTRU
- You can still "prove" security based on average case assumption: Breaking the above hash function is as hard as finding short vectors in a random lattice $\Lambda([\mathbf{A}^{(1)}|\ldots|\mathbf{A}^{(m/n)}])$
- ...but we know the function is broken: The underlying random lattice distribution is weak!
- Conclusion: Assuming that a problem is hard on average-case is a really tricky business!

Back to general lattices

- Finding short vectors in $\Lambda_q^\perp(\mathbf{A})$ when $\mathbf{A}$ is a random "block circulant" matrix is easy
- What about unstructured random $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$?

Question

Is $f_\mathbf{A}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ is random?

- Yes, provided SIVP/ADD/BDD are hard in the worst-case! [Ajtai96,...,MR04]
- We will give an oversimplified proof sketch, where $\mathbf{A} \in \mathbb{R}^{k \times n}$

Back to general lattices

- Finding short vectors in $\Lambda_q^{\perp}(\mathbf{A})$ when $\mathbf{A}$ is a random "block circulant" matrix is easy
- What about unstructured random $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$?

Question

Is $f_{\mathbf{A}}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ is random?

- Yes, provided SIVP/ADD/BDD are hard in the worst-case! [Ajtai96,...,MR04]
- We will give an oversimplified proof sketch, where $\mathbf{A} \in \mathbb{R}^{k \times n}$

Back to general lattices

- Finding short vectors in $\Lambda_q^\perp(\mathbf{A})$ when $\mathbf{A}$ is a random "block circulant" matrix is easy
- What about unstructured random $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$?

Question

Is $f_\mathbf{A}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ is random?

- Yes, provided SIVP/ADD/BDD are hard in the worst-case! [Ajtai96,...,MR04]
- We will give an oversimplified proof sketch, where $\mathbf{A} \in \mathbb{R}^{k \times n}$

Back to general lattices

- Finding short vectors in $\Lambda_q^\perp(\mathbf{A})$ when $\mathbf{A}$ is a random "block circulant" matrix is easy
- What about unstructured random $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$?

Question

Is $f_{\mathbf{A}}$ collision resistant when $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ is random?

- Yes, provided SIVP/ADD/BDD are hard in the worst-case! [Ajtai96,...,MR04]
- We will give an oversimplified proof sketch, where $\mathbf{A} \in \mathbb{R}^{k \times n}$

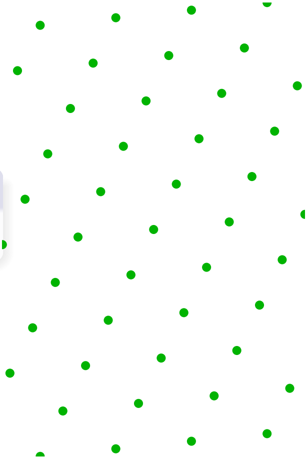
Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.



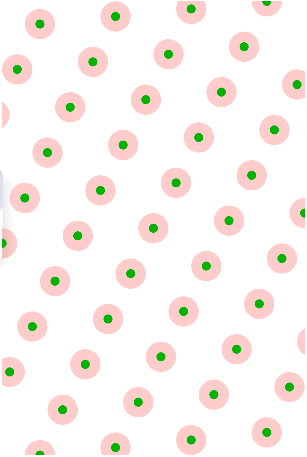
Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed? [MR]

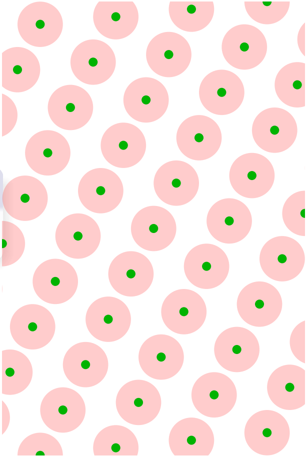
$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.



Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.



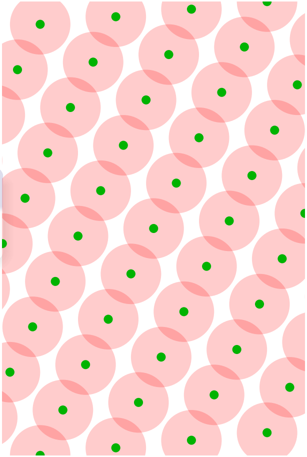
How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.

Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.



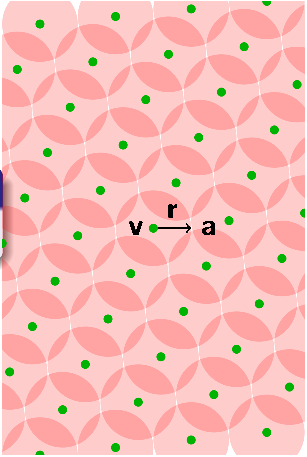
How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.

Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.



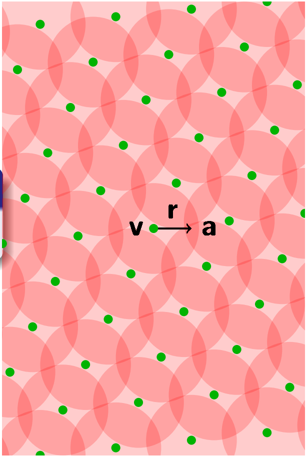
How much noise is needed? [MR]

$\|\mathbf{r}\| \le (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.

Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.



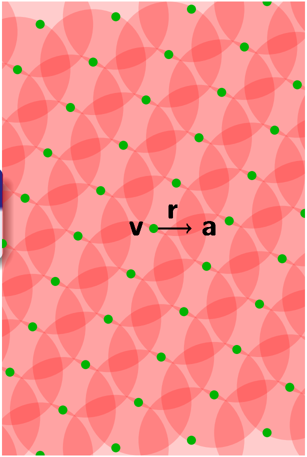
How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.

Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.



How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.

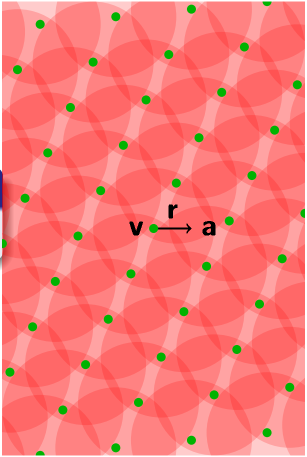
Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed? [MR]

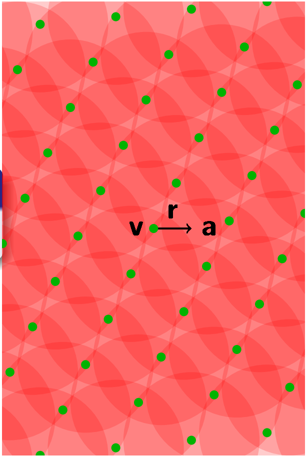
$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n / 2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.



Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.



How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n / 2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.

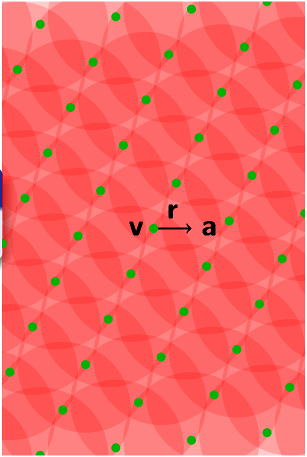
Blurring a lattice

Consider an arbitrary lattice, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n$ is uniformly distributed.



Security proof (sketch)

- Generate random points $\mathbf{a}_i = \mathbf{v}_i + \mathbf{r}_i$, where
    - $\mathbf{v}_i$ is a random lattice point
    - $\mathbf{r}_i$ is a random error vector of length $\|\mathbf{r}_i\| \approx \sqrt{n}\lambda_n$
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ is distributed almost uniformly at random in $\mathbb{R}^{n\times m}$, so
    - if we can break Ajtai's function $f_{\mathbf{A}}$, then
    - we can find a vector $\mathbf{z} \in \{-1, 0, 1\}^m$ such that

$$\sum(\mathbf{v}_i + \mathbf{r}_i)z_i = \sum \mathbf{a}_i z_i = \mathbf{0}$$

- Rearranging the terms yields a lattice vector

$$\sum \mathbf{v}_i z_i = -\sum \mathbf{r}_i z_i$$

of length at most $\|\sum \mathbf{r}_i x_i\| \approx \sqrt{n} \cdot \max \|\mathbf{r}_i\| \approx n \cdot \lambda_n$

Security proof (sketch)

- Generate random points $\mathbf{a}_i = \mathbf{v}_i + \mathbf{r}_i$, where
  - $\mathbf{v}_i$ is a random lattice point
  - $\mathbf{r}_i$ is a random error vector of length $\|\mathbf{r}_i\| \approx \sqrt{n}\lambda_n$
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ is distributed almost uniformly at random in $\mathbb{R}^{n \times m}$, so
  - if we can break Ajtai's function $f_{\mathbf{A}}$, then
  - we can find a vector $\mathbf{z} \in \{-1, 0, 1\}^m$ such that

$$\sum (\mathbf{v}_i + \mathbf{r}_i) z_i = \sum \mathbf{a}_i z_i = \mathbf{0}$$

- Rearranging the terms yields a lattice vector

$$\sum \mathbf{v}_i z_i = -\sum \mathbf{r}_i z_i$$

of length at most $\|\sum \mathbf{r}_i x_i\| \approx \sqrt{n} \cdot \max \|\mathbf{r}_i\| \approx n \cdot \lambda_n$

Security proof (sketch)

- Generate random points $\mathbf{a}_i = \mathbf{v}_i + \mathbf{r}_i$, where
  - $\mathbf{v}_i$ is a random lattice point
  - $\mathbf{r}_i$ is a random error vector of length $\|\mathbf{r}_i\| \approx \sqrt{n}\lambda_n$
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ is distributed almost uniformly at random in $\mathbb{R}^{n\times m}$, so
  - if we can break Ajtai's function $f_{\mathbf{A}}$, then
  - we can find a vector $\mathbf{z} \in \{-1, 0, 1\}^m$ such that

$$\sum (\mathbf{v}_i + \mathbf{r}_i) z_i = \sum \mathbf{a}_i z_i = \mathbf{0}$$

- Rearranging the terms yields a lattice vector

$$\sum \mathbf{v}_i z_i = -\sum \mathbf{r}_i z_i$$

of length at most $\|\sum \mathbf{r}_i x_i\| \approx \sqrt{n} \cdot \max \|\mathbf{r}_i\| \approx n \cdot \lambda_n$

Security proof (sketch)

- Generate random points $\mathbf{a}_i = \mathbf{v}_i + \mathbf{r}_i$, where
  - $\mathbf{v}_i$ is a random lattice point
  - $\mathbf{r}_i$ is a random error vector of length $\|\mathbf{r}_i\| \approx \sqrt{n}\lambda_n$
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$ is distributed almost uniformly at random in $\mathbb{R}^{n \times m}$, so
  - if we can break Ajtai's function $f_{\mathbf{A}}$, then
  - we can find a vector $\mathbf{z} \in \{-1, 0, 1\}^m$ such that

$$\sum (\mathbf{v}_i + \mathbf{r}_i) z_i = \sum \mathbf{a}_i z_i = \mathbf{0}$$

- Rearranging the terms yields a lattice vector

$$\sum \mathbf{v}_i z_i = -\sum \mathbf{r}_i z_i$$

of length at most $\|\sum \mathbf{r}_i x_i\| \approx \sqrt{n} \cdot \max \|\mathbf{r}_i\| \approx n \cdot \lambda_n$

What about efficiency

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
1	-4	-3	-8	6	-4	-9	-0	2	-6	-4	-5	3	-2	-7	-1
8	1	-4	-3	0	6	-4	-9	5	2	-6	-4	1	3	-2	-7
3	8	1	-4	9	0	6	-4	4	5	2	-6	7	1	3	-2
4	3	8	1	4	9	0	6	6	4	5	2	2	7	1	3

What about efficiency

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
1	-4	-3	-8	6	-4	-9	-0	2	-6	-4	-5	3	-2	-7	-1
8	1	-4	-3	0	6	-4	-9	5	2	-6	-4	1	3	-2	-7
3	8	1	-4	9	0	6	-4	4	5	2	-6	7	1	3	-2
4	3	8	1	4	9	0	6	6	4	5	2	2	7	1	3

Theorem (trivial)

Finding collisions on the average is at least as hard as finding short vectors in the corresponding random lattices

What about efficiency

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
1	-4	-3	-8	6	-4	-9	-0	2	-6	-4	-5	3	-2	-7	-1
8	1	-4	-3	0	6	-4	-9	5	2	-6	-4	1	3	-2	-7
3	8	1	-4	9	0	6	-4	4	5	2	-6	7	1	3	-2
4	3	8	1	4	9	0	6	6	4	5	2	2	7	1	3

Theorem (trivial)

Finding collisions on the average is at least as hard as finding short vectors in the corresponding random lattices

Theorem (LM'07)

Provably collision resistant, assuming the worst case hardness of approximating SVP and SIVP over ideal lattices.

Efficiency of anti-cyclic hashing

- Key size: $(m/n) \cdot n \log q = m \cdot \log q = \tilde{O}(n)$ bits
- Anti-cyclic matrix-vector multiplication can be computed in quasi-linear time $\tilde{O}(n)$ using FFT
- The resulting hash function can also be computed in $\tilde{O}(n)$ time
- For approximate choice of parameters, this can be very practical (SWIFFT [LMPR])
- The hash function is linear: $\mathbf{A}(\mathbf{x} + \mathbf{y}) = \mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{y}$
- We will see that this can be a feature rather than a weakness

Outline

1. Point Lattices
   - Computational Problems
   - The dual lattice
2. Lattice Cryptography
   - Average Case Hardness
   - Random Lattices
   - Cryptographic functions

Hard Random Lattices

Theorem (Ajtai,MR04)

$f_{\mathbf{A}}$ is collision resistant, under the assumption that SIVP is hard to approximate in the worst-case withing a factor $\gamma \approx n$.

Equivalently, ...

Theorem

If ADD is hard to approximate in the worst case within $\gamma \approx n$, then ADD is hard on average for input distribution $\Lambda_q^\perp(\mathbb{Z}_q^{n \times m})$.

Theorem (R05)

If ADD/SIVP is hard to approximate in the worst case within $\gamma \approx n$ even by quantum algorithms, then BDD is hard on average for input distribution $\Lambda_q^\perp(\mathbb{Z}_q^{n \times m})$.

Hard Random Lattices

Theorem (Ajtai,MR04)

$f_{\mathbf{A}}$ is collision resistant, under the assumption that SIVP is hard to approximate in the worst-case withing a factor $\gamma \approx n$.

Equivalently, ...

Theorem

If ADD is hard to approximate in the worst case within $\gamma \approx n$, then ADD is hard on average for input distribution $\Lambda_q^{\perp}(\mathbb{Z}_q^{n \times m})$.

Theorem (R05)

If ADD/SIVP is hard to approximate in the worst case within $\gamma \approx n$ even by quantum algorithms, then BDD is hard on average for input distribution $\Lambda_q^{\perp}(\mathbb{Z}_q^{n \times m})$.

Hard Random Lattices

Theorem (Ajtai,MR04)

$f_{\mathbf{A}}$ is collision resistant, under the assumption that SIVP is hard to approximate in the worst-case withing a factor $\gamma \approx n$.

Equivalently, ...

Theorem

If ADD is hard to approximate in the worst case within $\gamma \approx n$, then ADD is hard on average for input distribution $\Lambda_q^{\perp}(\mathbb{Z}_q^{n\times m})$.

Theorem (R05)

If ADD/SIVP is hard to approximate in the worst case within $\gamma \approx n$ even by quantum algorithms, then BDD is hard on average for input distribution $\Lambda_q^{\perp}(\mathbb{Z}_q^{n\times m})$.

One-time signatures

- OTS: diginal signature scheme that allows to sign a single message (faster than a full fledged signature scheme)
- Global parameters: q-ary lattice $\mathbf{A}$
- Secret key: short error vectors $\mathbf{S}$
- Public key: syndromes $\mathbf{P} = \mathbf{AS}$ (Hash of secret key under homomorphic hash function)
- Message: short vector $\mathbf{m}$
- Signature: $\sigma = \mathbf{Sm}$
- Verify: Check if σ is short and $\mathbf{Pm} = \mathbf{A}\sigma$

One-time signatures

- OTS: diginal signature scheme that allows to sign a single message (faster than a full fledged signature scheme)
- Global parameters: q-ary lattice $\mathbf{A}$
- Secret key: short error vectors $\mathbf{S}$
- Public key: syndromes $\mathbf{P} = \mathbf{AS}$ (Hash of secret key under homomorphic hash function)
- Message: short vector $\mathbf{m}$
- Signature: $\sigma = \mathbf{Sm}$
- Verify: Check if σ is short and $\mathbf{Pm} = \mathbf{A}\sigma$

One-time signatures

- OTS: diginal signature scheme that allows to sign a single message (faster than a full fledged signature scheme)
- Global parameters: q-ary lattice $\mathbf{A}$
- Secret key: short error vectors $\mathbf{S}$
- Public key: syndromes $\mathbf{P} = \mathbf{AS}$ (Hash of secret key under homomorphic hash function)
- Message: short vector $\mathbf{m}$
- Signature: $\sigma = \mathbf{Sm}$
- Verify: Check if σ is short and $\mathbf{Pm} = \mathbf{A}\sigma$

One-time signatures

- OTS: diginal signature scheme that allows to sign a single message (faster than a full fledged signature scheme)
- Global parameters: q-ary lattice $\mathbf{A}$
- Secret key: short error vectors $\mathbf{S}$
- Public key: syndromes $\mathbf{P} = \mathbf{AS}$ (Hash of secret key under homomorphic hash function)
- Message: short vector $\mathbf{m}$
- Signature: $\sigma = \mathbf{Sm}$
- Verify: Check if σ is short and $\mathbf{Pm} = \mathbf{A}\sigma$

One-time signatures

- OTS: diginal signature scheme that allows to sign a single message (faster than a full fledged signature scheme)
- Global parameters: q-ary lattice $\mathbf{A}$
- Secret key: short error vectors $\mathbf{S}$
- Public key: syndromes $\mathbf{P} = \mathbf{AS}$ (Hash of secret key under homomorphic hash function)
- Message: short vector $\mathbf{m}$
- Signature: $\sigma = \mathbf{Sm}$
- Verify: Check if σ is short and $\mathbf{Pm} = \mathbf{A}\sigma$

One-time signatures

- OTS: diginal signature scheme that allows to sign a single message (faster than a full fledged signature scheme)
- Global parameters: q-ary lattice $\mathbf{A}$
- Secret key: short error vectors $\mathbf{S}$
- Public key: syndromes $\mathbf{P} = \mathbf{AS}$ (Hash of secret key under homomorphic hash function)
- Message: short vector $\mathbf{m}$
- Signature: $\sigma = \mathbf{Sm}$
- Verify: Check if σ is short and $\mathbf{Pm} = \mathbf{A}\sigma$

One-time signatures

- OTS: diginal signature scheme that allows to sign a single message (faster than a full fledged signature scheme)
- Global parameters: q-ary lattice $\mathbf{A}$
- Secret key: short error vectors $\mathbf{S}$
- Public key: syndromes $\mathbf{P} = \mathbf{AS}$ (Hash of secret key under homomorphic hash function)
- Message: short vector $\mathbf{m}$
- Signature: $\sigma = \mathbf{Sm}$
- Verify: Check if σ is short and $\mathbf{Pm} = \mathbf{A}\sigma$

OTS security

Assume there is an attack to the one-time signature scheme. Then we can find collisions to hash function $f_{\mathbf{A}}$ as follows.

- Generate $\mathbf{A}$, $\mathbf{S}$, $\mathbf{P} = \mathbf{AS}$
- Sign $\sigma = \mathbf{Sm}$ as requested by attacker
- Attacker produces a forgery $(\mathbf{m}', \sigma')$
- $(\mathbf{Sm}', \sigma')$ is a collision: $\mathbf{ASm}' = \mathbf{Pm}' = \mathbf{A}\sigma'$

Note: Adversary cannot output $\sigma' = \mathbf{Sm}'$ because $\mathbf{A}, \mathbf{P}, \sigma$ do not reveal enough information about $\mathbf{S}$.

Note: This scheme [LM08] can be very efficient when implemented with ideal lattices.

OTS security

Assume there is an attack to the one-time signature scheme. Then we can find collisions to hash function $f_{\mathbf{A}}$ as follows.

- Generate $\mathbf{A}$, $\mathbf{S}$, $\mathbf{P} = \mathbf{AS}$
- Sign $\sigma = \mathbf{Sm}$ as requested by attacker
- Attacker produces a forgery $(\mathbf{m}', \sigma')$
- $(\mathbf{Sm}', \sigma')$ is a collision: $\mathbf{ASm}' = \mathbf{Pm}' = \mathbf{A}\sigma'$

Note: Adversary cannot output $\sigma' = \mathbf{Sm}'$ because $\mathbf{A}, \mathbf{P}, \sigma$ do not reveal enough information about $\mathbf{S}$.

Note: This scheme [LM08] can be very efficient when implemented with ideal lattices.

OTS security

Assume there is an attack to the one-time signature scheme. Then we can find collisions to hash function $f_{\mathbf{A}}$ as follows.

- Generate $\mathbf{A}$, $\mathbf{S}$, $\mathbf{P} = \mathbf{AS}$
- Sign $\sigma = \mathbf{Sm}$ as requested by attacker
- Attacker produces a forgery $(\mathbf{m}', \sigma')$
- $(\mathbf{Sm}', \sigma')$ is a collision: $\mathbf{ASm}' = \mathbf{Pm}' = \mathbf{A}\sigma'$

Note: Adversary cannot output $\sigma' = \mathbf{Sm}'$ because $\mathbf{A}, \mathbf{P}, \sigma$ do not reveal enough information about $\mathbf{S}$.

Note: This scheme [LM08] can be very efficient when implemented with ideal lattices.

OTS security

Assume there is an attack to the one-time signature scheme. Then we can find collisions to hash function $f_{\mathbf{A}}$ as follows.

- Generate $\mathbf{A}$, $\mathbf{S}$, $\mathbf{P} = \mathbf{AS}$
- Sign $\sigma = \mathbf{Sm}$ as requested by attacker
- Attacker produces a forgery $(\mathbf{m}', \sigma')$
- $(\mathbf{Sm}', \sigma')$ is a collision: $\mathbf{ASm}' = \mathbf{Pm}' = \mathbf{A}\sigma'$

Note: Adversary cannot output $\sigma' = \mathbf{Sm}'$ because $\mathbf{A}, \mathbf{P}, \sigma$ do not reveal enough information about $\mathbf{S}$.

Note: This scheme [LM08] can be very efficient when implemented with ideal lattices.

OTS security

Assume there is an attack to the one-time signature scheme. Then we can find collisions to hash function $f_{\mathbf{A}}$ as follows.

- Generate $\mathbf{A}$, $\mathbf{S}$, $\mathbf{P} = \mathbf{AS}$
- Sign $\sigma = \mathbf{Sm}$ as requested by attacker
- Attacker produces a forgery $(\mathbf{m}', \sigma')$
- $(\mathbf{Sm}', \sigma')$ is a collision: $\mathbf{ASm}' = \mathbf{Pm}' = \mathbf{A}\sigma'$

Note: Adversary cannot output $\sigma' = \mathbf{Sm}'$ because $\mathbf{A}, \mathbf{P}, \sigma$ do not reveal enough information about $\mathbf{S}$.

Note: This scheme [LM08] can be very efficient when implemented with ideal lattices.

OTS security

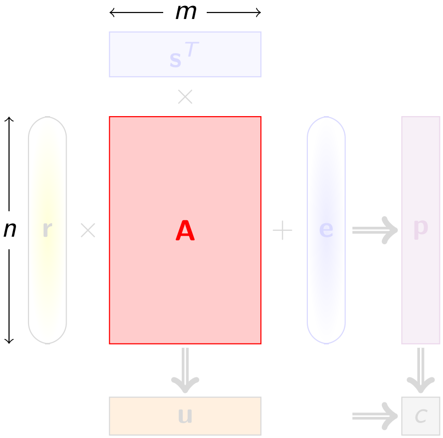
Assume there is an attack to the one-time signature scheme. Then we can find collisions to hash function $f_{\mathbf{A}}$ as follows.

- Generate $\mathbf{A}$, $\mathbf{S}$, $\mathbf{P} = \mathbf{AS}$
- Sign $\sigma = \mathbf{Sm}$ as requested by attacker
- Attacker produces a forgery $(\mathbf{m}', \sigma')$
- $(\mathbf{Sm}', \sigma')$ is a collision: $\mathbf{ASm}' = \mathbf{Pm}' = \mathbf{A}\sigma'$

Note: Adversary cannot output $\sigma' = \mathbf{Sm}'$ because $\mathbf{A}, \mathbf{P}, \sigma$ do not reveal enough information about $\mathbf{S}$.

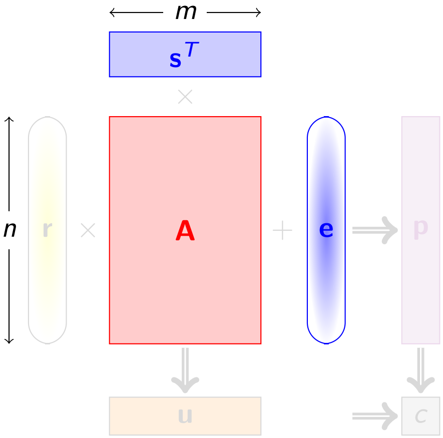
Note: This scheme [LM08] can be very efficient when implemented with ideal lattices.

Regev (LWE) cryptosystem



- Parameters: $m, n, q \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$
- Public key: $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- Encrypt$_{\mathbf{p}}(m;(\mathbf{r}))$:
$$\begin{aligned} \mathbf{u} &= \mathbf{r}^T \mathbf{A} \\ c &= \mathbf{r}^T \mathbf{p} + m - r_0 \end{aligned}$$
- Decrypt$_{\mathbf{s}}(\mathbf{u}, c) =$ $c - \mathbf{u} \cdot \mathbf{s} \approx m.$

Regev (LWE) cryptosystem

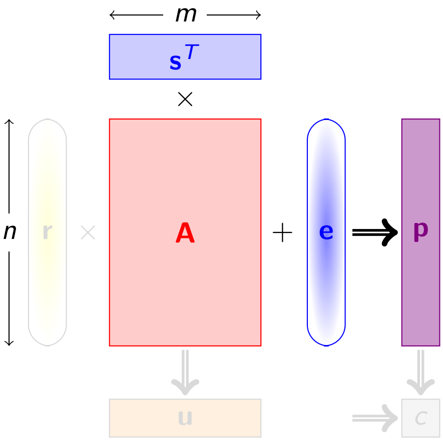


- Parameters: $m, n, q \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$
- Public key: $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- $\text{Encrypt}_{\mathbf{p}}(m;(\mathbf{r}))$:

$$\begin{aligned} \mathbf{u} &= \mathbf{r}^T\mathbf{A} \\ c &= \mathbf{r}^T\mathbf{p} + m - r_0 \end{aligned}$$

- $\text{Decrypt}_{\mathbf{s}}(\mathbf{u},c) =$ $c - \mathbf{u} \cdot \mathbf{s} \approx m.$

Regev (LWE) cryptosystem

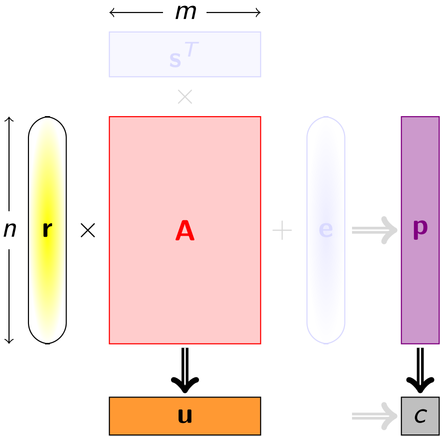


- Parameters: $m, n, q \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$
- Public key: $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- Encrypt$_{\mathbf{p}}(m;(\mathbf{r}))$:

$$\begin{aligned} \mathbf{u} &= \mathbf{r}^T\mathbf{A} \\ c &= \mathbf{r}^T\mathbf{p} + m - r_0 \end{aligned}$$

- Decrypt$_{\mathbf{s}}(\mathbf{u}, c) = c - \mathbf{u} \cdot \mathbf{s} \approx m$.

Regev (LWE) cryptosystem

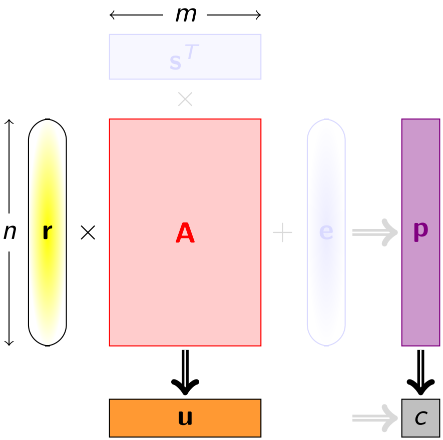


- Parameters:
  $m, n, q \in \mathbb{Z}, \mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$
- Public key:
  $\mathbf{p} = \mathbf{As} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- Encrypt$_{\mathbf{p}}(m;(\mathbf{r}))$:

$$\begin{aligned} \mathbf{u} &= \mathbf{r}^T\mathbf{A} \\ c &= \mathbf{r}^T\mathbf{p} + m - r_0 \end{aligned}$$

- Decrypt$_{\mathbf{s}}(\mathbf{u}, c) =$
  $c - \mathbf{u} \cdot \mathbf{s} \approx m.$

Regev (LWE) cryptosystem

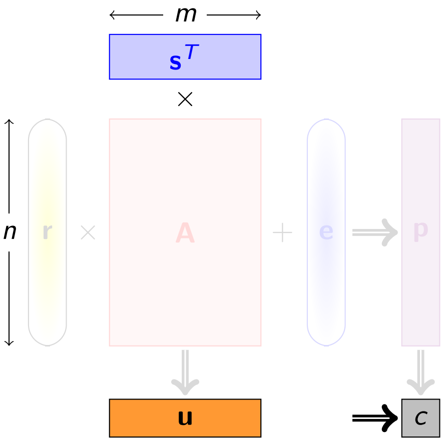


- Parameters: $m, n, q \in \mathbb{Z}, \mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$
- Public key: $\mathbf{p} = \mathbf{As} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- Encrypt$_\mathbf{p}(m;(\mathbf{r}))$:

$$\begin{aligned} \mathbf{u} &= \mathbf{r}^T \mathbf{A} \\ c &= \mathbf{r}^T \mathbf{p} + m - r_0 \end{aligned}$$

- Decrypt$_\mathbf{s}(\mathbf{u}, c) =$ $c - \mathbf{u} \cdot \mathbf{s} \approx m$.

Regev (LWE) cryptosystem

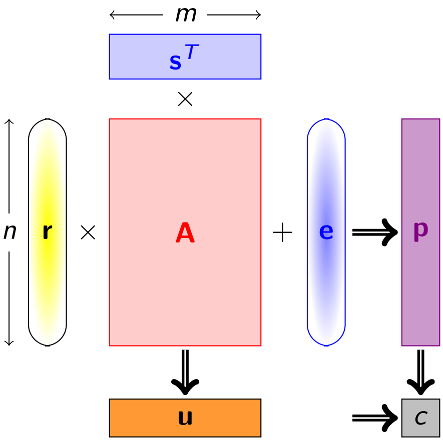


- Parameters:
  $m, n, q \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$
- Public key:
  $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- $\mathrm{Encrypt}_{\mathbf{p}}(m;(\mathbf{r}))$:

$$\begin{aligned} \mathbf{u} &= \mathbf{r}^T\mathbf{A} \\ c &= \mathbf{r}^T\mathbf{p} + m - r_0 \end{aligned}$$

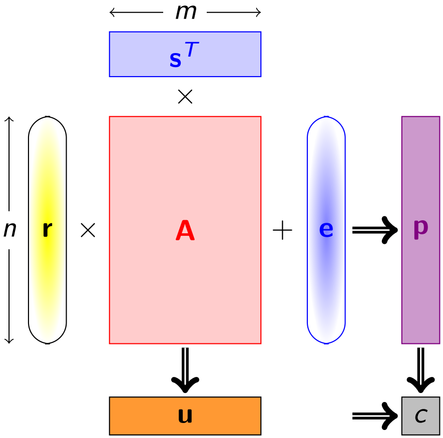
- $\mathrm{Decrypt}_{\mathbf{s}}(\mathbf{u},c) =$
  $c - \mathbf{u} \cdot \mathbf{s} \approx m.$

The geometry of LWE encryption



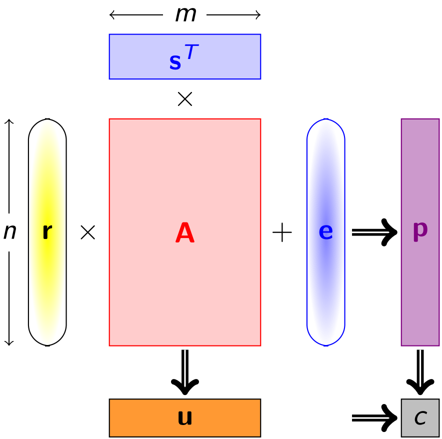
- Public key:
  $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- $[\mathbf{A} \mid \mathbf{p}]$: random q-ary lattice with a planted short vector $\mathbf{e}$
- Encryption:
  $(\mathbf{u}, c) = [\mathbf{A}|\mathbf{p}]^T\mathbf{r}$ is the syndrome of $\mathbf{r} + \Lambda_q^\perp([\mathbf{A}|\mathbf{p}])$
- Decryption: use short dual vector $\mathbf{e}$ to solve BDD problem

The geometry of LWE encryption



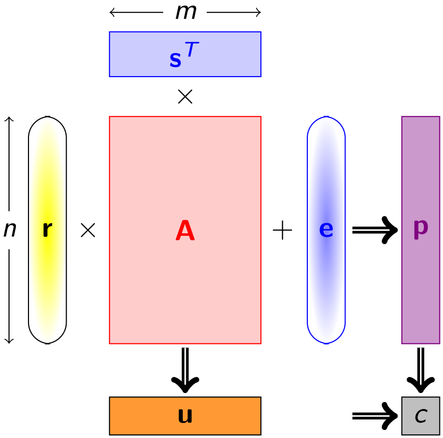
- Public key:
  $\mathbf{p} = \mathbf{As} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- $[\mathbf{A} \mid \mathbf{p}]$: random q-ary lattice with a planted short vector $\mathbf{e}$
- Encryption:
  $(\mathbf{u}, c) = [\mathbf{A}|\mathbf{p}]^T\mathbf{r}$ is the syndrome of $\mathbf{r} + \Lambda_q^\perp([\mathbf{A}|\mathbf{p}])$
- Decryption: use short dual vector $\mathbf{e}$ to solve BDD problem

The geometry of LWE encryption



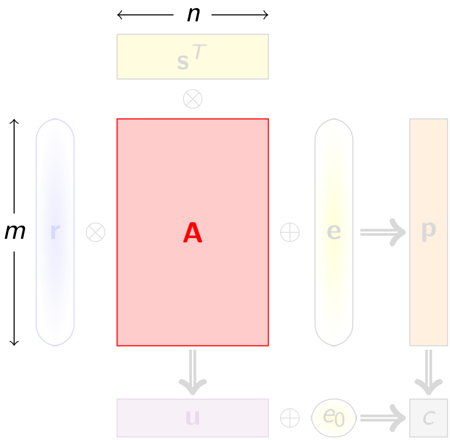
- Public key:
  $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- $[\mathbf{A} \mid \mathbf{p}]$: random q-ary lattice with a planted short vector $\mathbf{e}$
- Encryption:
  $(\mathbf{u}, c) = [\mathbf{A}|\mathbf{p}]^T\mathbf{r}$ is the syndrome of $\mathbf{r} + \Lambda_q^\perp([\mathbf{A}|\mathbf{p}])$
- Decryption: use short dual vector $\mathbf{e}$ to solve BDD problem

The geometry of LWE encryption



- Public key: $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \approx_c \mathbb{Z}_q^m$
- $[\mathbf{A} \mid \mathbf{p}]$: random q-ary lattice with a planted short vector $\mathbf{e}$
- Encryption: $(\mathbf{u}, c) = [\mathbf{A}|\mathbf{p}]^T\mathbf{r}$ is the syndrome of $\mathbf{r} + \Lambda_q^\perp([\mathbf{A}|\mathbf{p}])$
- Decryption: use short dual vector $\mathbf{e}$ to solve BDD problem

GPV (dual LWE) cryptosystem

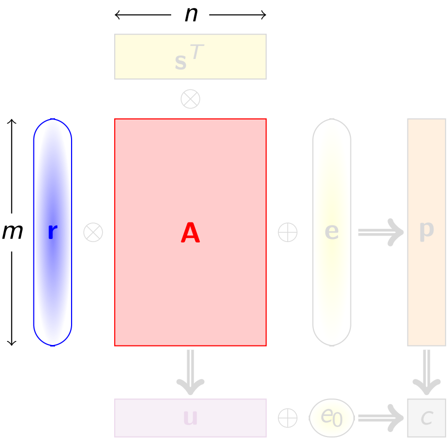


- Parameters: $m, n, q \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{r} \in \mathcal{E}^m$
- Public key: $\mathbf{u} = \mathbf{r}^T\mathbf{A} \approx_s \mathbb{Z}_q^m$
- $\text{Encrypt}_{\mathbf{u}}(m;\mathbf{e})$:

$$\begin{aligned} \mathbf{p} &= \mathbf{A}\mathbf{s} + \mathbf{e} \\ c &= \mathbf{u} \cdot \mathbf{s} + e_0 + m \end{aligned}$$

- $\text{Decrypt}_{\mathbf{r}}(\mathbf{p}, c) = c - \mathbf{r}^T\mathbf{p} \approx m.$

GPV (dual LWE) cryptosystem

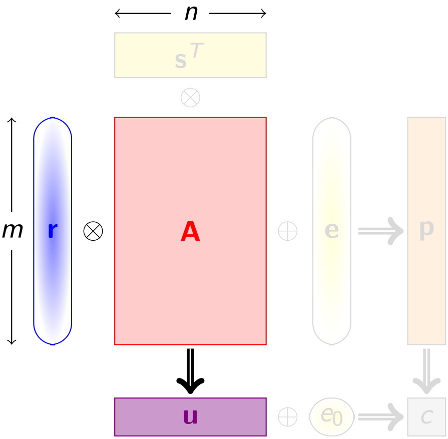


- Parameters: $m, n, q \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$
- Secret key: $\mathbf{r} \in \mathcal{E}^m$
- Public key: $\mathbf{u} = \mathbf{r}^T\mathbf{A} \approx_s \mathbb{Z}_q^m$
- Encrypt$_{\mathbf{u}}$(m;$\mathbf{e}$):

$$\begin{aligned}\mathbf{p} &= \mathbf{A}\mathbf{s} + \mathbf{e}\\ c &= \mathbf{u}\cdot\mathbf{s} + e_0 + m\end{aligned}$$

- Decrypt$_{\mathbf{r}}$($\mathbf{p}$,c) = $c - \mathbf{r}^T\mathbf{p} \approx m$.

GPV (dual LWE) cryptosystem

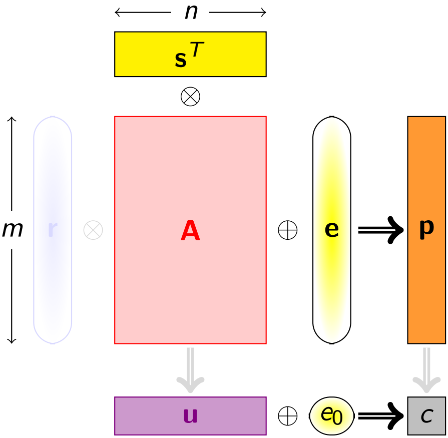


- Parameters: $m, n, q \in \mathbb{Z}, \mathbf{A} \in \mathbb{Z}_q^{m\times n}$
- Secret key: $\mathbf{r} \in \mathcal{E}^m$
- Public key: $\mathbf{u} = \mathbf{r}^T\mathbf{A} \approx_s \mathbb{Z}_q^m$
- Encrypt$_{\mathbf{u}}(m;\mathbf{e})$:

$$\begin{aligned} \mathbf{p} &= \mathbf{A}\mathbf{s} + \mathbf{e} \\ c &= \mathbf{u}\cdot\mathbf{s} + e_0 + m \end{aligned}$$

- Decrypt$_{\mathbf{r}}(\mathbf{p}, c) =$ $c - \mathbf{r}^T\mathbf{p} \approx m$.

GPV (dual LWE) cryptosystem

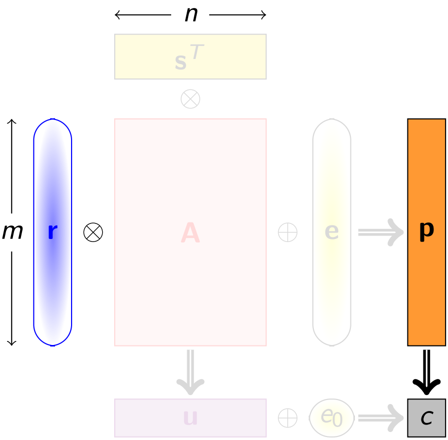


- Parameters: $m, n, q \in \mathbb{Z}, \mathbf{A} \in \mathbb{Z}_q^{m\times n}$
- Secret key: $\mathbf{r} \in \mathcal{E}^m$
- Public key: $\mathbf{u} = \mathbf{r}^T\mathbf{A} \approx_s \mathbb{Z}_q^m$
- Encrypt$_{\mathbf{u}}(m;\mathbf{e})$:

$$\begin{aligned} \mathbf{p} &= \mathbf{As} + \mathbf{e} \\ c &= \mathbf{u}\cdot\mathbf{s} + e_0 + m \end{aligned}$$

- Decrypt$_{\mathbf{r}}(\mathbf{p},c) = c - \mathbf{r}^T\mathbf{p} \approx m$.

GPV (dual LWE) cryptosystem

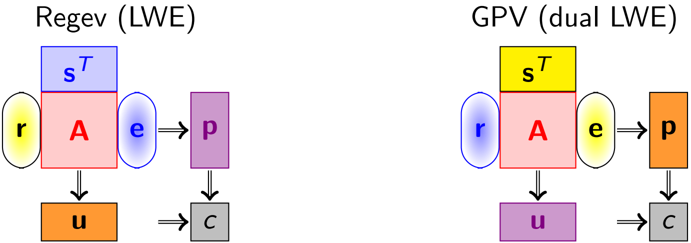


- Parameters: $m, n, q \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Secret key: $\mathbf{r} \in \mathcal{E}^m$
- Public key: $\mathbf{u} = \mathbf{r}^T\mathbf{A} \approx_s \mathbb{Z}_q^m$
- Encrypt$_{\mathbf{u}}(m;\mathbf{e})$:

$$\begin{aligned} \mathbf{p} &= \mathbf{As} + \mathbf{e} \\ c &= \mathbf{u} \cdot \mathbf{s} + e_0 + m \end{aligned}$$

- Decrypt$_{\mathbf{r}}(\mathbf{p},c) = c - \mathbf{r}^T\mathbf{p} \approx m$.

Comparing Regev and GPV encryption

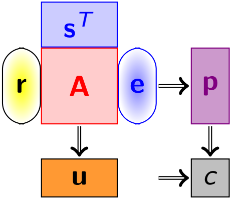


Regev and GPV cryptosystems use the same mathematical objects $\mathbf{A}, \mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{p}, \mathbf{u}, c$, but operate on them in different roles:

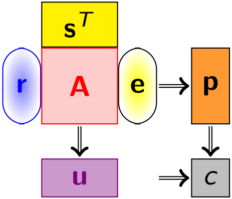
Public key generation	$\Longleftrightarrow$	Encryption
Secret key	$\Longleftrightarrow$	Encryption randomness
Public key	$\Longleftrightarrow$	Ciphertext

Comparing Regev and GPV encryption

Regev (LWE)



GPV (dual LWE)



Regev and GPV cryptosystems use the same mathematical objects $\mathbf{A}, \mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{p}, \mathbf{u}, c$, but operate on them in different roles:

Public key generation	$\Longleftrightarrow$	Encryption
Secret key	$\Longleftrightarrow$	Encryption randomness
Public key	$\Longleftrightarrow$	Ciphertext

Naive interpretation

- The schemes are syntactically similar: Regev and GPV cryptosystems operate on the same mathematical objects $\mathbf{A}, \mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{p}, \mathbf{u}, c$.
- The scheme are semantically different:

Common parameters	$\mathbf{A}$	$\Longleftrightarrow$	$\mathbf{A}$	Common parameters
secret key	$\mathbf{s}, \mathbf{e}$	$\Longleftrightarrow$	$\mathbf{s}, \mathbf{e}$	encryption randomness
encryption randomness	$\mathbf{r}$	$\Longleftrightarrow$	$\mathbf{r}$	secret key
public key	$\mathbf{p}$	$\Longleftrightarrow$	$\mathbf{p}$	ciphertext
ciphertext	$\mathbf{u}$	$\Longleftrightarrow$	$\mathbf{u}$	public key

Naive interpretation

- The schemes are syntactically similar: Regev and GPV cryptosystems operate on the same mathematical objects $\mathbf{A}, \mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{p}, \mathbf{u}, c$.
- The scheme are semantically different:

Common parameters	$\mathbf{A}$	$\Longleftrightarrow$	$\mathbf{A}$	Common parameters
secret key	$\mathbf{s}, \mathbf{e}$	$\Longleftrightarrow$	$\mathbf{s}, \mathbf{e}$	encryption randomness
encryption randomness	$\mathbf{r}$	$\Longleftrightarrow$	$\mathbf{r}$	secret key
public key	$\mathbf{p}$	$\Longleftrightarrow$	$\mathbf{p}$	ciphertext
ciphertext	$\mathbf{u}$	$\Longleftrightarrow$	$\mathbf{u}$	public key

The true answer: Lattices and Duality

- The schemes are syntactically different: The symbols $\mathbf{A}, \mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{p}, \mathbf{u}, c$ in Regev and GPV cryptosystems represent different mathematical objects
- The two schemes are semantically equivalent:

Common parameters	$\mathbf{A}$	$\Longleftrightarrow$	$\mathbf{A}'$	Common parameters
secret key	$\mathbf{s}, \mathbf{e}$	$\Longleftrightarrow$	$\mathbf{r}'$	secret key
encryption randomness	$\mathbf{r}$	$\Longleftrightarrow$	$\mathbf{s}', \mathbf{e}'$	encryption randomness
public key	$\mathbf{p}$	$\Longleftrightarrow$	$\mathbf{u}'$	public key
ciphertext	$\mathbf{u}$	$\Longleftrightarrow$	$\mathbf{p}'$	ciphertext

The true answer: Lattices and Duality

- The schemes are syntactically different: The symbols $\mathbf{A}, \mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{p}, \mathbf{u}, c$ in Regev and GPV cryptosystems represent different mathematical objects
- The two schemes are semantically equivalent:

Common parameters	$\mathbf{A}$	$\Longleftrightarrow$	$\mathbf{A}'$	Common parameters
secret key	$\mathbf{s}, \mathbf{e}$	$\Longleftrightarrow$	$\mathbf{r}'$	secret key
encryption randomness	$\mathbf{r}$	$\Longleftrightarrow$	$\mathbf{s}', \mathbf{e}'$	encryption randomness
public key	$\mathbf{p}$	$\Longleftrightarrow$	$\mathbf{u}'$	public key
ciphertext	$\mathbf{u}$	$\Longleftrightarrow$	$\mathbf{p}'$	ciphertext

Trapdoor functions

Theorem (A99,AP09,MP11)

There is an algorithm to efficiently generate a random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *together with a short basis* $\mathbf{S} \in \mathbb{Z}^{m \times m}$ *of* $\Lambda_q^\perp(\mathbf{A})$.

Trapdoor function:

- Inverting $f_{\mathbf{A}}$ is a BDD problem
- BDD can be solved with a short dual basis
- $\mathbf{S}$ can be used as an inversion trapdoor

Injective trapdoor functions can be used for the construction of a wide range of other more complex cryptographic primitives.

Trapdoor functions

Theorem (A99,AP09,MP11)

There is an algorithm to efficiently generate a random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *together with a short basis* $\mathbf{S} \in \mathbb{Z}^{m \times m}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$.

Trapdoor function:

- Inverting $f_{\mathbf{A}}$ is a BDD problem
- BDD can be solved with a short dual basis
- $\mathbf{S}$ can be used as an inversion trapdoor

Injective trapdoor functions can be used for the construction of a wide range of other more complex cryptographic primitives.

Conclusion

- Lattice cryptography allows to build a wide range of many other cryptographic primitives (Hierarchical identity based encryption, Fully homomorphic encryption, and much more)
- It has great potential for fast implementation due to simple operations and high parallelizability
- Most primitives can be described and explained in terms of a handful of basic geometric concepts
- Everything that can be done with number theoretic scheme can be done with lattice crypography as well
- Currently the only method known to build fully homomorphic encryption
- Not quite ready for use in practice, but moving fast in that direction
- Open problems: concrete efficiency, security evaluation, etc.