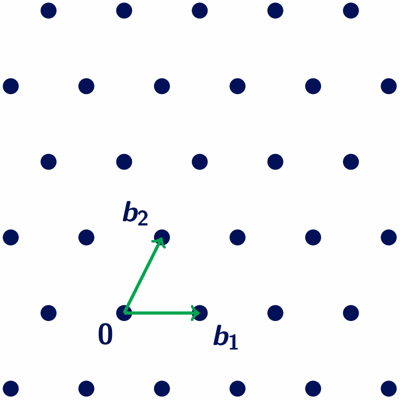
# The Lattice Isomorphism Problem

## algorithms and invariants

Wessel van Woerden (Université de Bordeaux, IMB, Inria).

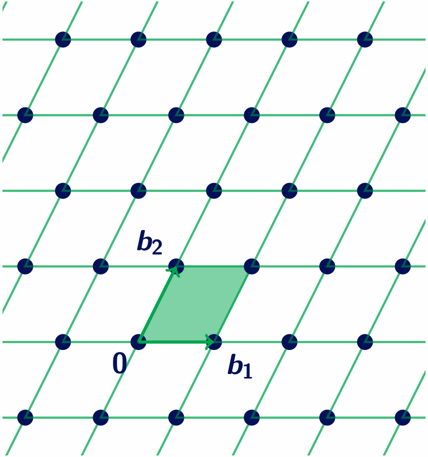



Lattice

$\mathbb{R}$-linearly independent $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$

$\mathcal{L}(B) := \{\sum_i x_i b_i : x \in \mathbb{Z}^n\} \subset \mathbb{R}^n$,

basis $B$, gram matrix $G := B^\top B$




Lattice

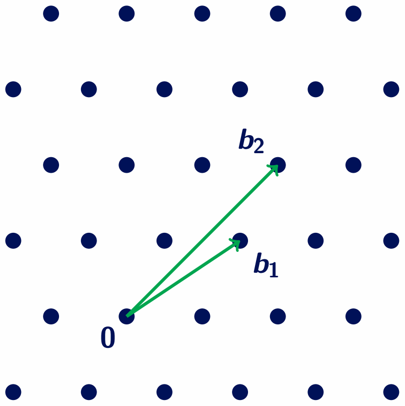
$\mathbb{R}$-linearly independent $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$

$$\mathcal{L}(B) := \{\sum_i x_i b_i : x \in \mathbb{Z}^n\} \subset \mathbb{R}^n,$$

basis $B$, gram matrix $G := B^\top B$

Lattice volume

$$\det(\mathcal{L}) := \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) = |\det(B)|$$




Lattice

$\mathbb{R}$-linearly independent $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$

$$\mathcal{L}(B) := \{\sum_i x_i b_i : x \in \mathbb{Z}^n\} \subset \mathbb{R}^n,$$

basis $B$, gram matrix $G := B^\top B$

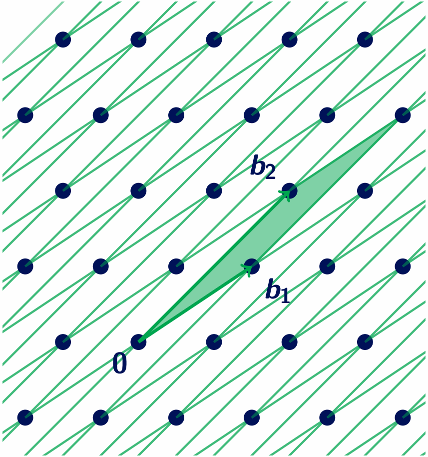
Lattice volume

$$\det(\mathcal{L}) := \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) = |\det(B)|$$

Infinitely many distinct bases

$$B' = B \cdot U, \quad G' = U^\top G U,$$

for $U \in \mathcal{GL}_n(\mathbb{Z})$.




Lattice

$\mathbb{R}$-linearly independent $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$

$$\mathcal{L}(B) := \{\textstyle\sum_i x_i b_i : x \in \mathbb{Z}^n\} \subset \mathbb{R}^n,$$

basis $B$, gram matrix $G := B^\top B$

Lattice volume

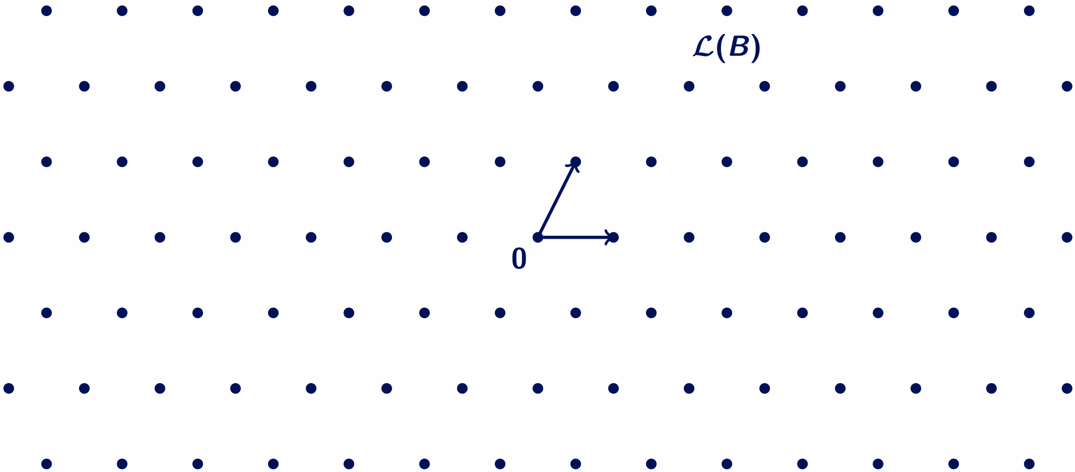
$$\mathbf{det}(\mathcal{L}) := \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) = |\,\mathbf{det}(B)|$$

Infinitely many distinct bases

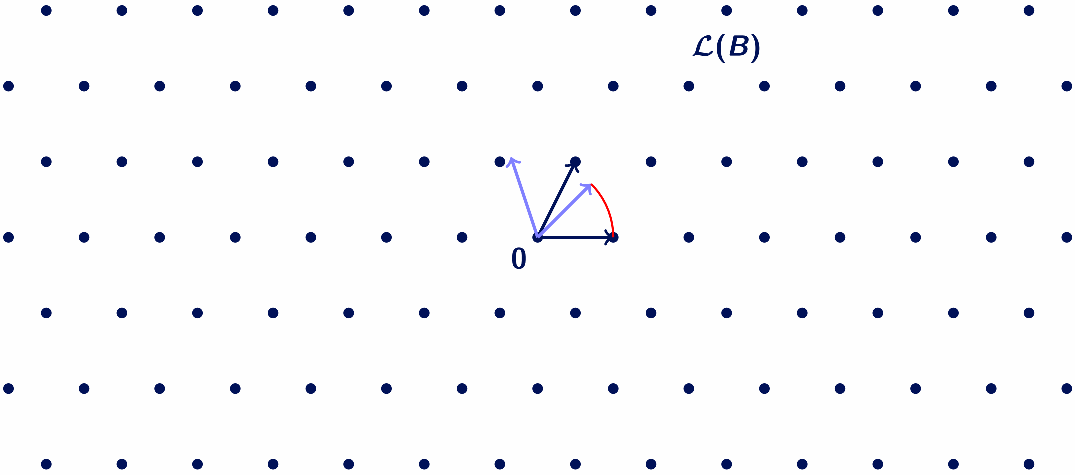
$$B' = B \cdot U, \quad G' = U^\top G U,$$

for $U \in \mathcal{GL}_n(\mathbb{Z})$.

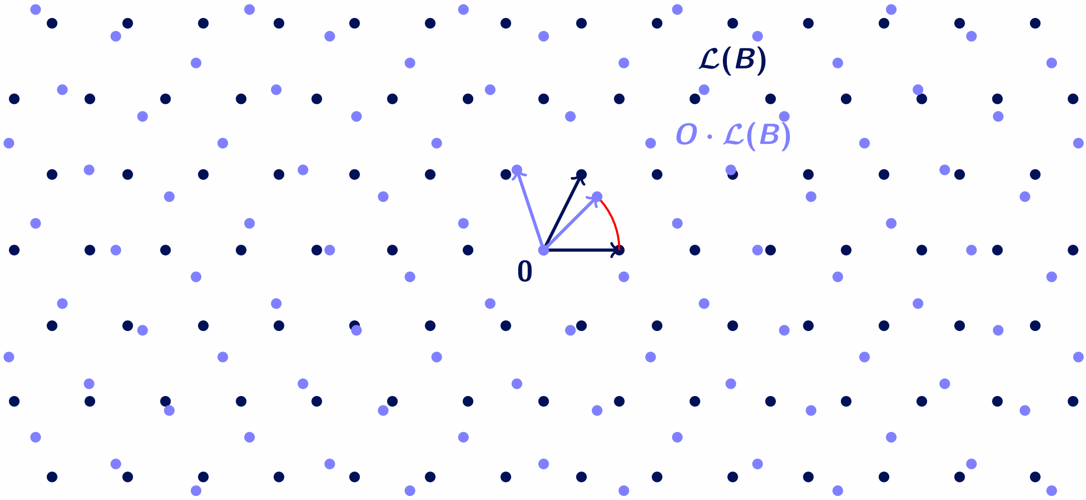
# Lattice Isomorphism Problem (LIP)



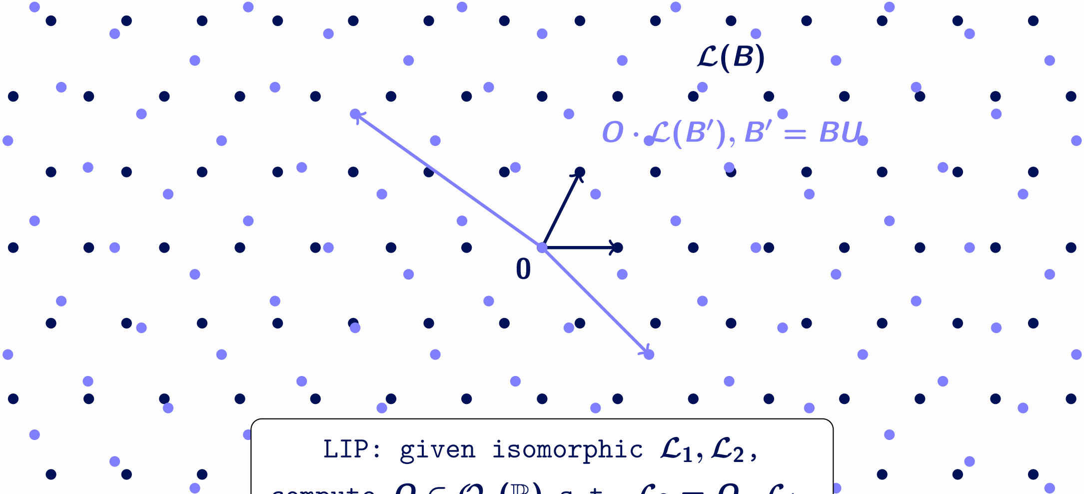
# Lattice Isomorphism Problem (LIP)



# Lattice Isomorphism Problem (LIP)

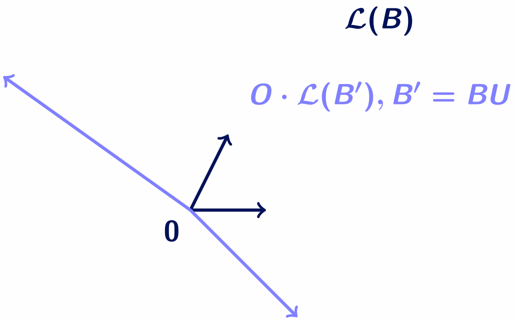


# Lattice Isomorphism Problem (LIP)




LIP: given isomorphic $\mathcal{L}_1, \mathcal{L}_2$,
compute $O \in \mathcal{O}_n(\mathbb{R})$ s.t. $\mathcal{L}_2 = O \cdot \mathcal{L}_1$.

# Lattice Isomorphism Problem (LIP)




LIP: given isomorphic $\mathcal{L}_1, \mathcal{L}_2$,
compute $O \in \mathcal{O}_n(\mathbb{R})$ s.t. $\mathcal{L}_2 = O \cdot \mathcal{L}_1$.

# Lattice Isomorphism Problem

$$\mathcal{L}(B_1) \cong \mathcal{L}(B_2)$$

$$\iff$$

$$O \cdot \mathcal{L}(B_1) = \mathcal{L}(B_2) \qquad \text{for some } O \in O_d(\mathbb{R})$$

$$\iff$$

$$O \cdot B_1 \cdot U = B_2 \qquad \text{for some } O \in O_d(\mathbb{R}), U \in \mathrm{GL}_d(\mathbb{Z})$$

# Lattice Isomorphism Problem

$$\mathcal{L}(B_1) \cong \mathcal{L}(B_2)$$

$$\Longleftrightarrow$$

$$O \cdot \mathcal{L}(B_1) = \mathcal{L}(B_2) \qquad \text{for some } O \in O_d(\mathbb{R})$$

$$\Longleftrightarrow$$

$$O \cdot B_1 \cdot U = B_2 \qquad \text{for some } O \in O_d(\mathbb{R}), U \in \mathrm{GL}_d(\mathbb{Z})$$

- If either $O$ or $U$ is trivial: linear algebra.

# Lattice Isomorphism Problem

$$\mathcal{L}(B_1) \cong \mathcal{L}(B_2)$$

$$\iff$$

$$O \cdot \mathcal{L}(B_1) = \mathcal{L}(B_2) \quad \text{for some } O \in \mathcal{O}_d(\mathbb{R})$$

$$\iff$$

$$O \cdot B_1 \cdot U = B_2 \quad \text{for some } O \in \mathcal{O}_d(\mathbb{R}), U \in \mathrm{GL}_d(\mathbb{Z})$$

$$\iff$$

$$U^t B_1^t B_1 U = \underbrace{B_2^t B_2}_{\text{gram matrix}} \quad \text{for some } U \in \mathrm{GL}_d(\mathbb{Z})$$

- If either $O$ or $U$ is trivial: linear algebra.
- Use $O^t O = I$ to remove the orthonormal transformation.

# Lattice Isomorphism Problem

$$\mathcal{L}(B_1) \cong \mathcal{L}(B_2)$$

$$\Longleftrightarrow$$

$$O \cdot \mathcal{L}(B_1) = \mathcal{L}(B_2) \qquad \text{for some } O \in O_d(\mathbb{R})$$

$$\Longleftrightarrow$$

$$O \cdot B_1 \cdot U = B_2 \qquad \text{for some } O \in O_d(\mathbb{R}), U \in \mathrm{GL}_d(\mathbb{Z})$$

$$\Longleftrightarrow$$

$$U^t B_1^t B_1 U = \underbrace{B_2^t B_2}_{\text{gram matrix}} \qquad \text{for some } U \in \mathrm{GL}_d(\mathbb{Z})$$

- If either $O$ or $U$ is trivial: linear algebra.
- Use $O^t O = I$ to remove the orthonormal transformation.
- We restrict to integer or rational gram matrices $G := B^\top B$.

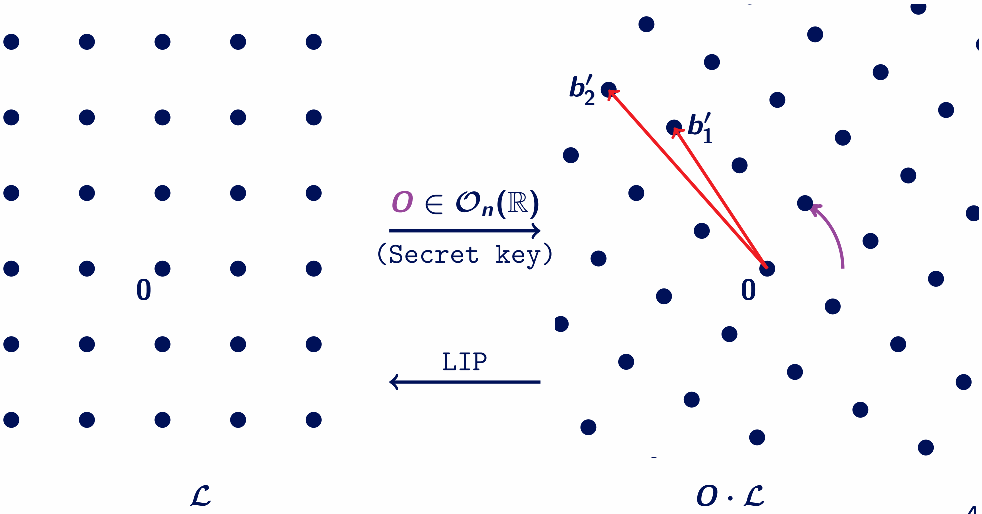
# Lattice Isomorphism Problem

$$\mathcal{L}(B_1) \cong \mathcal{L}(B_2)$$
$$\Longleftrightarrow$$
$$O \cdot \mathcal{L}(B_1) = \mathcal{L}(B_2) \qquad \text{for some } O \in O_d(\mathbb{R})$$
$$\Longleftrightarrow$$
$$O \cdot B_1 \cdot U = B_2 \qquad \text{for some } O \in O_d(\mathbb{R}), U \in \mathrm{GL}_d(\mathbb{Z})$$
$$\Longleftrightarrow$$
$$U^t B_1^t B_1 U = \underbrace{B_2^t B_2}_{\text{gram matrix}} \qquad \text{for some } U \in \mathrm{GL}_d(\mathbb{Z})$$

- If either $O$ or $U$ is trivial: linear algebra.
- Use $O^t O = I$ to remove the orthonormal transformation.
- We restrict to integer or rational gram matrices $G := B^\top B$.
- Solution unique up to $\mathrm{Aut}(\mathcal{L}) = \{O \in \mathcal{O}_n(\mathbb{R}) : O \cdot \mathcal{L} = \mathcal{L}\}$.

# Encryption scheme from LIP (informal)

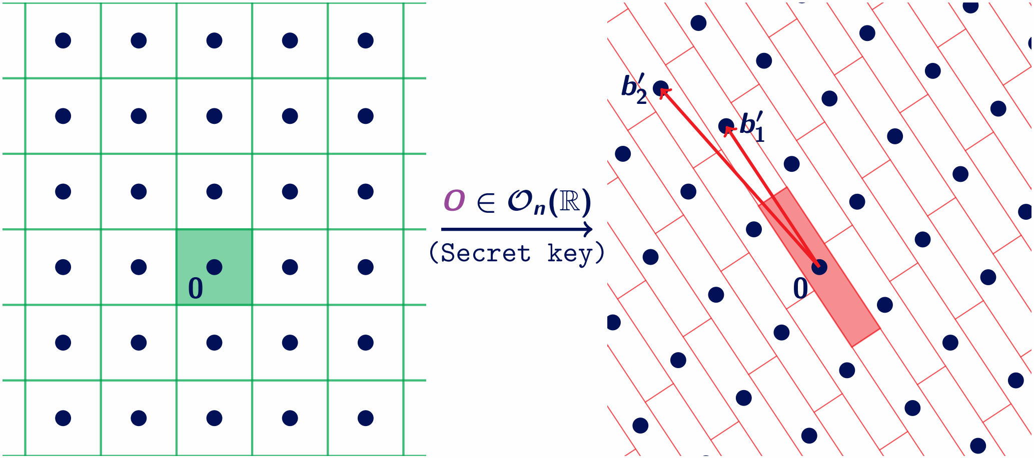
Decodable lattice | Bad basis of rotation



# Encryption scheme from LIP (informal)

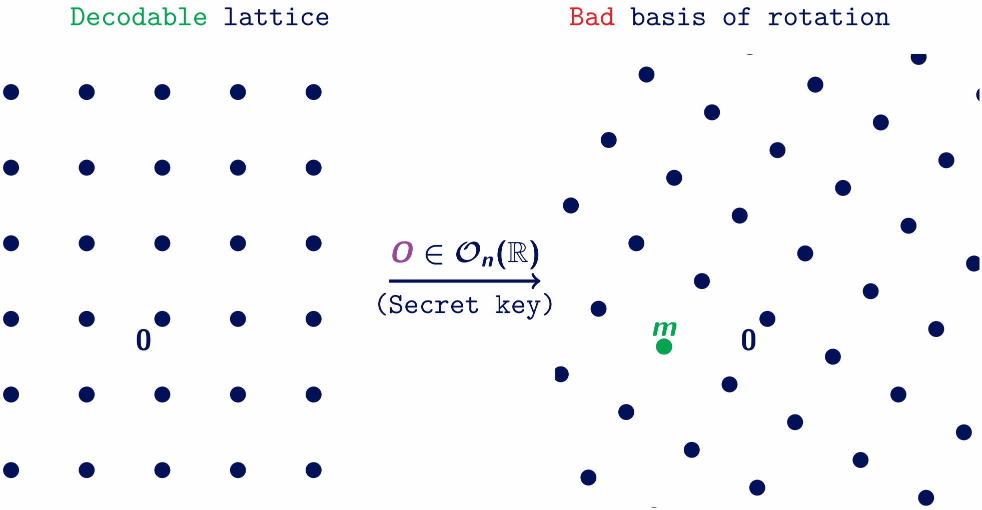
Decodable lattice

Bad basis of rotation

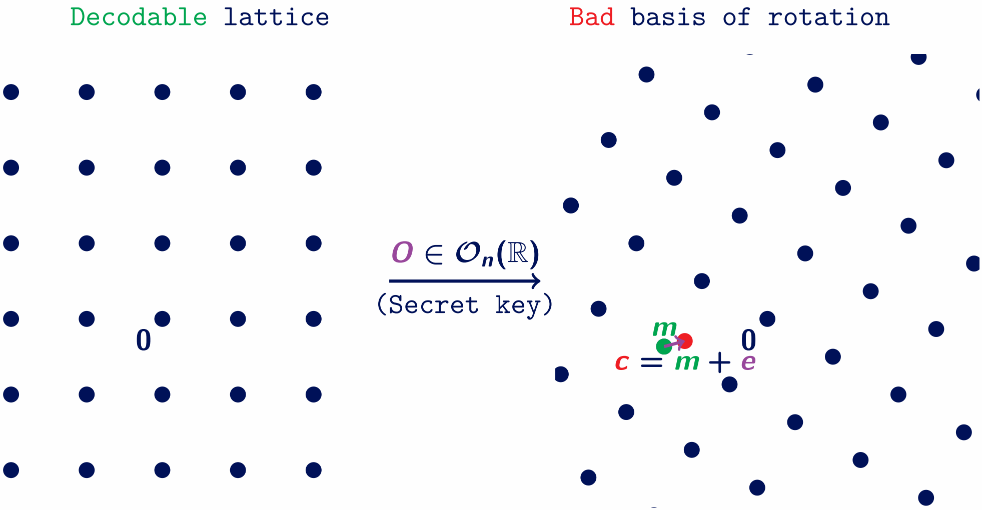



Hides (decoding) structure of $\mathcal{L}$

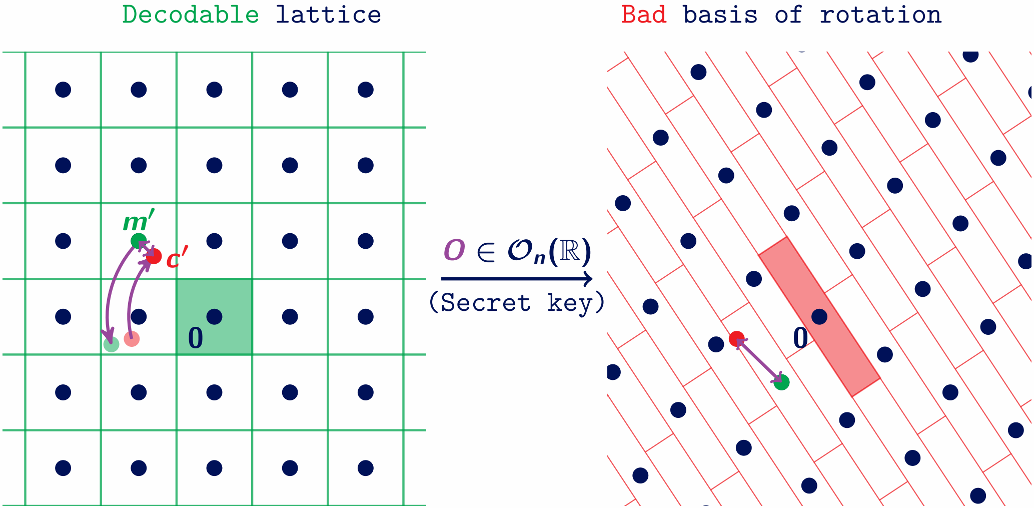
# Encryption scheme from LIP (informal)



# Encryption scheme from LIP (informal)




Encrypt by adding a small error




Decrypt using decoding algorithm

# Cryptography from LIP

- LIP as a new hardness assumption

# Cryptography from LIP

- LIP as a new hardness assumption

**Ducas & vW: On LIP, QFs, Remarkable Lattices, and Cryptography**

Use LIP to hide a remarkable lattice:

- Identification, Encryption and Signature scheme

# Cryptography from LIP

- LIP as a new hardness assumption

**Ducas & vW: On LIP, QFs, Remarkable Lattices, and Cryptography**

Use LIP to hide a remarkable lattice:

- Identification, Encryption and Signature scheme

**Bennett et al.: Just how hard are rotations of $\mathbb{Z}^n$?**

- Encryption scheme based on LIP on $\mathbb{Z}^n$,

# Cryptography from LIP

- LIP as a new hardness assumption

**Ducas & vW: On LIP, QFs, Remarkable Lattices, and Cryptography**

Use LIP to hide a remarkable lattice:

- Identification, Encryption and Signature scheme

**Bennett et al.: Just how hard are rotations of $\mathbb{Z}^n$?**

- Encryption scheme based on LIP on $\mathbb{Z}^n$,

**Ducas et al.: HAWK scheme**

Efficient signature scheme based on module-LIP on $\mathbb{Z}^n$

- submitted to NIST call for additional signatures

- Several others works using LIP appeared recently

# Algorithms for solving LIP

# Main strategy for solving LIP

Goal: given isomorphic $\mathcal{L}, \mathcal{L}' \subset \mathbb{R}^n$, compute $O \in \mathcal{O}_n(\mathbb{R})$ s.t. $\mathcal{L}' = O \cdot \mathcal{L}$.

# Main strategy for solving LIP

Goal: given isomorphic $\mathcal{L}, \mathcal{L}' \subset \mathbb{R}^n$, compute $O \in \mathcal{O}_n(\mathbb{R})$ s.t. $\mathcal{L}' = O \cdot \mathcal{L}$.

Idea: isometries preserve lengths and inner products

$\Longrightarrow$ short(est) vectors map to short(est) vectors

# Main strategy for solving LIP

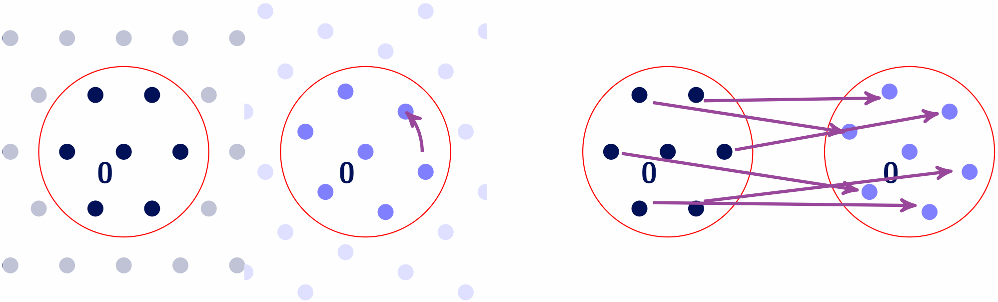
Goal: given isomorphic $\mathcal{L}, \mathcal{L}' \subset \mathbb{R}^n$, compute $O \in \mathcal{O}_n(\mathbb{R})$ s.t. $\mathcal{L}' = O \cdot \mathcal{L}$.

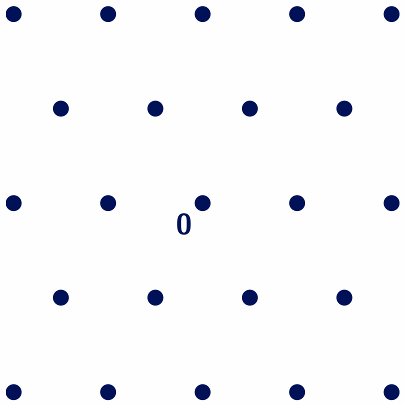
Idea: isometries preserve lengths and inner products

$\implies$ short(est) vectors map to short(est) vectors

Step 1:
compute short vectors

Step 2:
compute isometries between them



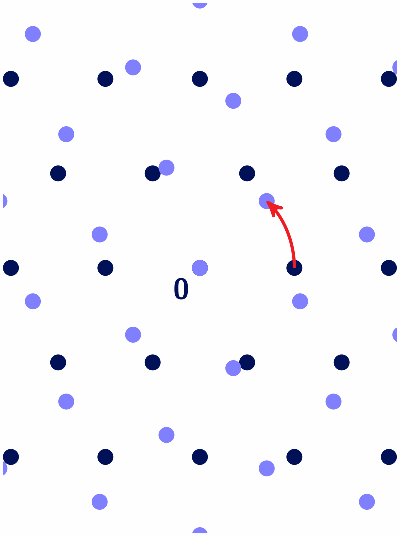



**Definition: characteristic vector set**

$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L}) \ \forall O \in \mathcal{O}_n(\mathbb{R})$.

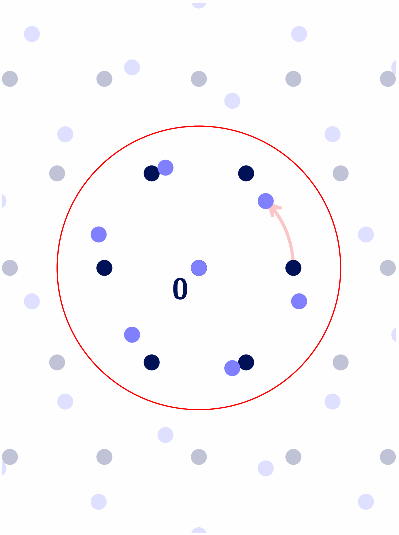



**Definition: characteristic vector set**

$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L})\ \forall O \in \mathcal{O}_n(\mathbb{R})$.




**Definition: characteristic vector set**

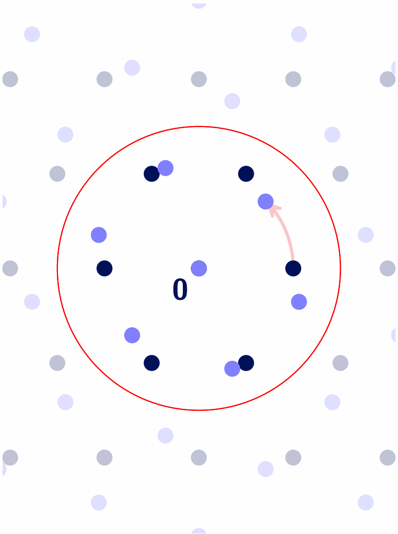
$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L})\ \forall O \in \mathcal{O}_n(\mathbb{R})$.

Example:

- Property **(2)** is satisfied e.g. by $\mathtt{Min}(\mathcal{L}, \lambda) := \{x \in \mathcal{L} : \|\mathcal{L}\| \leq \lambda\}$.




**Definition: characteristic vector set**

$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

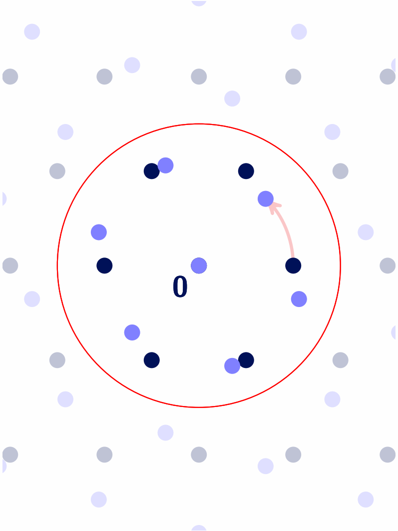
(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L})\ \forall O \in \mathcal{O}_n(\mathbb{R})$.

Example:

- Property **(2)** is satisfied e.g. by $\mathtt{Min}(\mathcal{L}, \lambda) := \{x \in \mathcal{L} : \|\mathcal{L}\| \leq \lambda\}$.
- $\mathcal{V}_{\mathtt{ms}}(\mathcal{L}) := \mathtt{Min}(\mathcal{L}, \lambda_{\min}(\mathcal{L}))$ with $\lambda_{\min}(\mathcal{L})$ minimal s.t. **(1)** is satisfied.

# Characteristic Vector Set




**Definition: characteristic vector set**

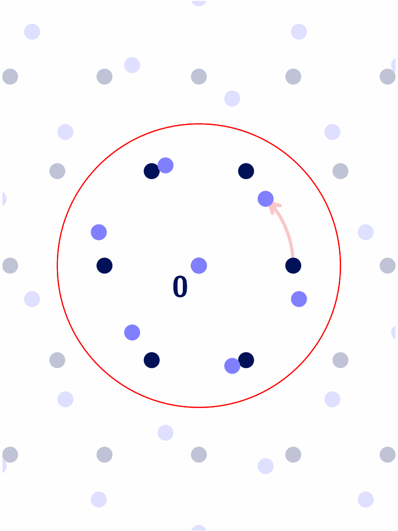
$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L})\ \forall O \in \mathcal{O}_n(\mathbb{R})$.

Example:

- Property **(2)** is satisfied e.g. by $\mathtt{Min}(\mathcal{L}, \lambda) := \{x \in \mathcal{L} : \|\mathcal{L}\| \leq \lambda\}$.
- $\mathcal{V}_{\mathtt{ms}}(\mathcal{L}) := \mathtt{Min}(\mathcal{L}, \lambda_{\min}(\mathcal{L}))$ with $\lambda_{\min}(\mathcal{L})$ minimal s.t. **(1)** is satisfied.
- $\mathcal{V}_{\mathtt{vor}}(\mathcal{L}) :=$ {Voronoi relevant vectors of $\mathcal{L}$}.




**Definition: characteristic vector set**

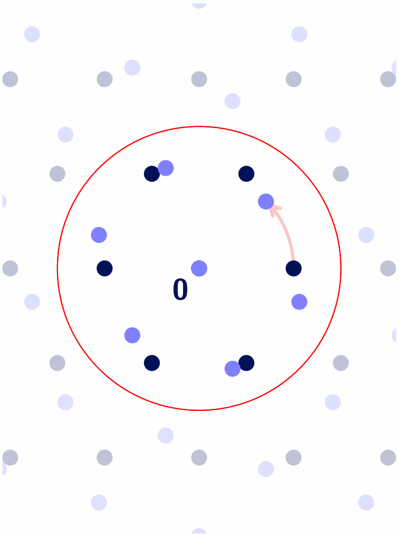
$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L}) \; \forall O \in \mathcal{O}_n(\mathbb{R})$.

Example:

- Property **(2)** is satisfied e.g. by $\mathtt{Min}(\mathcal{L}, \lambda) := \{x \in \mathcal{L} : \|\mathcal{L}\| \leq \lambda\}$.
- $\mathcal{V}_{\mathtt{ms}}(\mathcal{L}) := \mathtt{Min}(\mathcal{L}, \lambda_{\min}(\mathcal{L}))$ with $\lambda_{\min}(\mathcal{L})$ minimal s.t. **(1)** is satisfied.
- $\mathcal{V}_{\mathtt{vor}}(\mathcal{L}) :=$ {Voronoi relevant vectors of $\mathcal{L}$}.
- Complexity: $2^{O(n)}$ time and memory.




**Definition: characteristic vector set**

$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

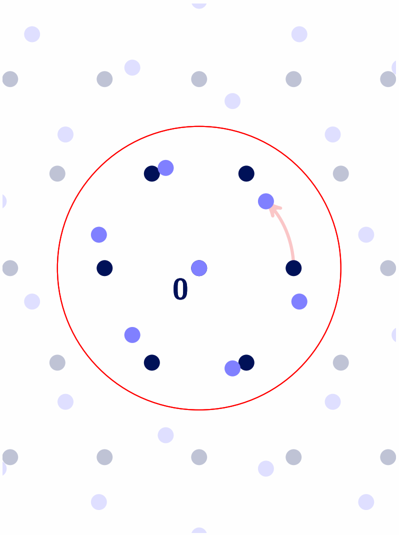
(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L})\ \forall O \in \mathcal{O}_n(\mathbb{R})$.

- Can be used as a proxy:

$$\mathcal{L}_2 = O \cdot \mathcal{L}_1$$

$$\Longleftrightarrow$$

$$\mathcal{V}(\mathcal{L}_2) \underbrace{=}_{\text{as a Set}} O \cdot \mathcal{V}(\mathcal{L}_1)$$




**Definition: characteristic vector set**

$\mathcal{V} : \mathcal{L} \mapsto \mathcal{V}(\mathcal{L}) \subset \mathcal{L}$ is a CVS if

(1) $\mathcal{V}(\mathcal{L})$ generates $\mathcal{L}$.

(2) $\mathcal{V}(O \cdot \mathcal{L}) = O \cdot \mathcal{V}(\mathcal{L})\ \forall O \in \mathcal{O}_n(\mathbb{R})$.

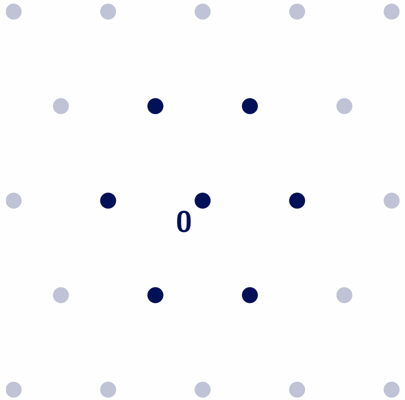
- Can be used as a proxy:

$$\mathcal{L}_2 = O \cdot \mathcal{L}_1$$

$$\Longleftrightarrow$$

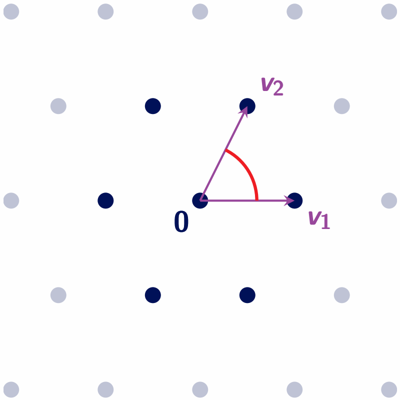
$$\mathcal{V}(\mathcal{L}_2) \underbrace{=}_{\text{as a Set}} O \cdot \mathcal{V}(\mathcal{L}_1)$$

- Goal: find a ~~linear~~ isometry $O : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$.

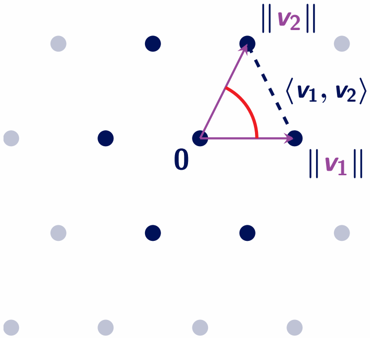



- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.

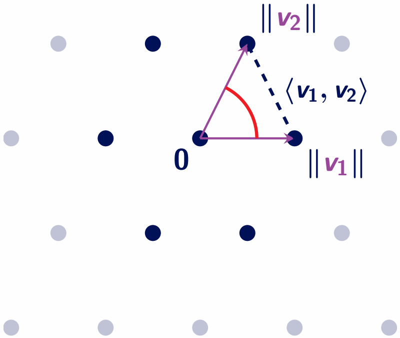
# LIP via Graph Isomorphism



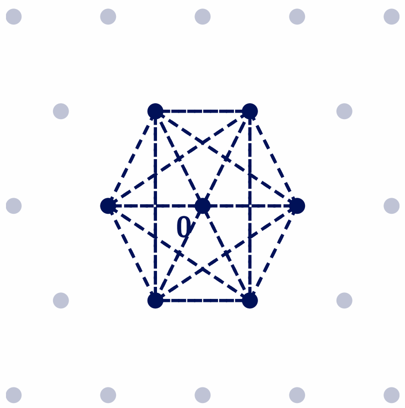

- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.



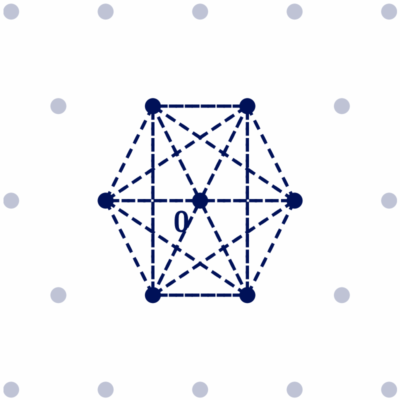

- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.




- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.
- Idea: this condition is sufficient.



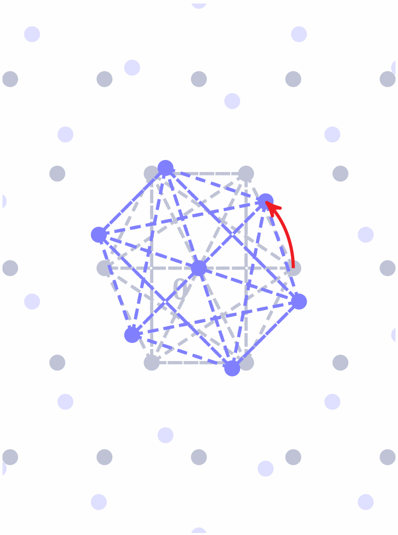

- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.
- Idea: this condition is sufficient.
- Let $G_{\mathcal{V}(\mathcal{L})} = (V, \omega)$ be a complete weighted graph with:
  - $V := \mathcal{V}(\mathcal{L}) = \{v_1, \ldots, v_N\}$
  - $\omega(v_i, v_j) := \langle v_i, v_j \rangle \quad \forall v_i, v_j \in V$.




- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.
- Idea: this condition is sufficient.
- Let $G_{\mathcal{V}(\mathcal{L})} = (V, \omega)$ be a complete weighted graph with:
  - $V := \mathcal{V}(\mathcal{L}) = \{v_1, \ldots, v_N\}$
  - $\omega(v_i, v_j) := \langle v_i, v_j \rangle \quad \forall v_i, v_j \in V$.
- Then:

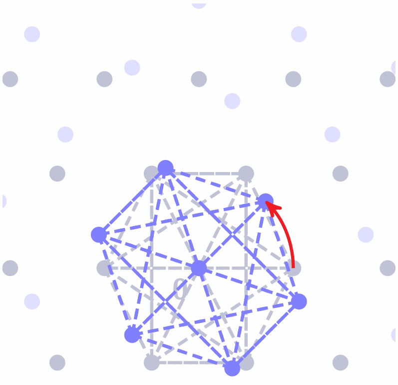
$$\mathcal{L}_1 \cong \mathcal{L}_2 \iff G_{\mathcal{V}(\mathcal{L}_1)} \cong G_{\mathcal{V}(\mathcal{L}_2)}$$

# LIP via Graph Isomorphism



- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.
- Idea: this condition is sufficient.
- Let $G_{\mathcal{V}(\mathcal{L})} = (V, \omega)$ be a complete weighted graph with:
  - $V := \mathcal{V}(\mathcal{L}) = \{v_1, \ldots, v_N\}$
  - $\omega(v_i, v_j) := \langle v_i, v_j \rangle \quad \forall v_i, v_j \in V$.
- Then:

$$\mathcal{L}_1 \cong \mathcal{L}_2 \iff G_{\mathcal{V}(\mathcal{L}_1)} \cong G_{\mathcal{V}(\mathcal{L}_2)}$$



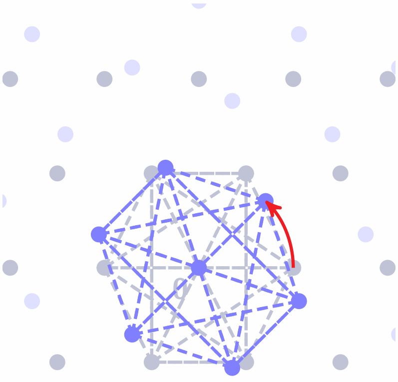
Time complexity:
$\exp(\log(|\mathcal{V}(\mathcal{L})|)^{O(1)})$
$= O(\exp(n^{O(1)}))$

- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.
- Idea: this condition is sufficient.
- Let $G_{\mathcal{V}(\mathcal{L})} = (V, \omega)$ be a complete weighted graph with:
  - $V := \mathcal{V}(\mathcal{L}) = \{v_1, \ldots, v_N\}$
  - $\omega(v_i, v_j) := \langle v_i, v_j \rangle \quad \forall v_i, v_j \in V$.
- Then:

$$\mathcal{L}_1 \cong \mathcal{L}_2 \iff G_{\mathcal{V}(\mathcal{L}_1)} \cong G_{\mathcal{V}(\mathcal{L}_2)}$$

- Problem: possibly $|\mathcal{V}(\mathcal{L})| \geq 2^{\Omega(n)}$.

# LIP via Graph Isomorphism



Time complexity:
$$\exp(\log(|\mathcal{V}(\mathcal{L})|)^{O(1)}) = O(\exp(n^{O(1)}))$$

- isometry $O : \mathcal{V}(\mathcal{L}_2) \to \mathcal{V}(\mathcal{L}_1)$ preserves pairwise inner products.
- Idea: this condition is sufficient.
- Let $G_{\mathcal{V}(\mathcal{L})} = (V, \omega)$ be a complete weighted graph with:
  - $V := \mathcal{V}(\mathcal{L}) = \{v_1, \ldots, v_N\}$
  - $\omega(v_i, v_j) := \langle v_i, v_j \rangle \quad \forall v_i, v_j \in V$.
- Then:

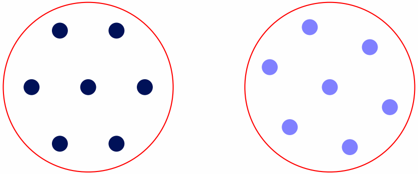
$$\mathcal{L}_1 \cong \mathcal{L}_2 \iff G_{\mathcal{V}(\mathcal{L}_1)} \cong G_{\mathcal{V}(\mathcal{L}_2)}$$

- Problem: possibly $|\mathcal{V}(\mathcal{L})| \geq 2^{\Omega(n)}$.
- Canonical graph labeling algorithms $\implies$ canonical form for LIP.

(Dutour Sikirić-Haensch-Voight-vW 2020)

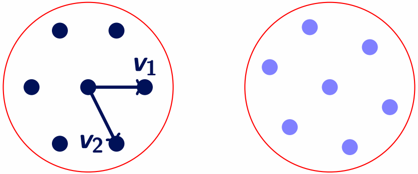
# Plesken-Souvignier (1997)

- Idea: linear isometry $f : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$ is fully determined by image on $n$ independent vectors.



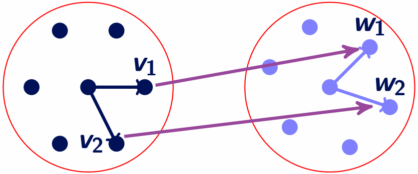
# Plesken-Souvignier (1997)

- Idea: linear isometry $f : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$ is fully determined by image on $n$ independent vectors.
- Let $v_1, \ldots, v_n \in \mathcal{V}(\mathcal{L}_1)$ be independent.



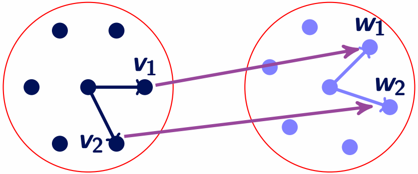
# Plesken-Souvignier (1997)

- Idea: linear isometry $f : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$ is fully determined by image on $n$ independent vectors.
- Let $v_1, \ldots, v_n \in \mathcal{V}(\mathcal{L}_1)$ be independent.
- Backtrack search to determine (compatible) images $f(v_1), \ldots, f(v_n) \in \mathcal{V}(\mathcal{L}_2)$.



# Plesken-Souvignier (1997)

- Idea: linear isometry $f : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$ is fully determined by image on $n$ independent vectors.
- Let $v_1, \ldots, v_n \in \mathcal{V}(\mathcal{L}_1)$ be independent.
- Backtrack search to determine (compatible) images $f(v_1), \ldots, f(v_n) \in \mathcal{V}(\mathcal{L}_2)$.




- Prune search tree: once $f(v_i) = w_i$ for $i = 1, \ldots, k$, then

$$\langle f(v_{k+1}), w_i \rangle = \langle f(v_{k+1}), f(v_i) \rangle = \langle v_{k+1}, v \rangle,$$

so possible images of $v_{k+1}$ are limited.

# Plesken-Souvignier (1997)

- Idea: linear isometry $f : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$ is fully determined by image on $n$ independent vectors.
- Let $v_1, \ldots, v_n \in \mathcal{V}(\mathcal{L}_1)$ be independent.
- Backtrack search to determine (compatible) images $f(v_1), \ldots, f(v_n) \in \mathcal{V}(\mathcal{L}_2)$.
- Prune search tree: once $f(v_i) = w_i$ for $i = 1, \ldots, k$, then

$$\langle f(v_{k+1}), w_i \rangle = \langle f(v_{k+1}), f(v_i) \rangle = \langle v_{k+1}, v \rangle,$$

  so possible images of $v_{k+1}$ are limited.
- Use more invariants to limit search-tree.

- Idea: linear isometry $f : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$ is fully determined by image on $n$ independent vectors.
- Let $v_1, \ldots, v_n \in \mathcal{V}(\mathcal{L}_1)$ be independent.
- Backtrack search to determine (compatible) images $f(v_1), \ldots, f(v_n) \in \mathcal{V}(\mathcal{L}_2)$.
- Prune search tree: once $f(v_i) = w_i$ for $i = 1, \ldots, k$, then

$$\langle f(v_{k+1}), w_i \rangle = \langle f(v_{k+1}), f(v_i) \rangle = \langle v_{k+1}, v \rangle,$$

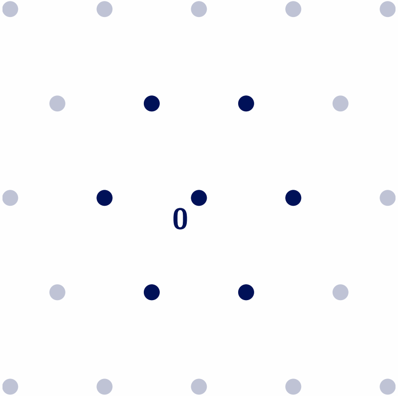
  so possible images of $v_{k+1}$ are limited.
- Use more invariants to limit search-tree.
- Good in practice, but tree can be as large as $\mathcal{O}\left(n! \cdot \binom{|\mathcal{V}(\mathcal{L})|}{n}\right)$.

# Plesken-Souvignier (1997)

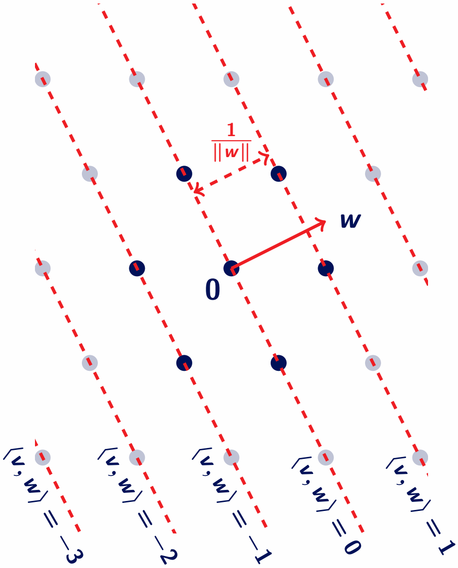
- Idea: linear isometry $f : \mathcal{V}(\mathcal{L}_1) \to \mathcal{V}(\mathcal{L}_2)$ is fully determined by image on $n$ independent vectors.
- Let $v_1, \ldots, v_n \in \mathcal{V}(\mathcal{L}_1)$ be independent.
- Backtrack search to determine (compatible) images $f(v_1), \ldots, f(v_n) \in \mathcal{V}(\mathcal{L}_2)$.
- Prune search tree: once $f(v_i) = w_i$ for $i = 1, \ldots, k$, then

$$\langle f(v_{k+1}), w_i \rangle = \langle f(v_{k+1}), f(v_i) \rangle = \langle v_{k+1}, v \rangle,$$

  so possible images of $v_{k+1}$ are limited.
- Use more invariants to limit search-tree.
- Good in practice, but tree can be as large as $\mathcal{O}\left(n! \cdot \binom{|\mathcal{V}(\mathcal{L})|}{n}\right)$.
- If $|\mathcal{V}(\mathcal{L})| = 2^{\Omega(n)}$ then $2^{O(n^2)}$ in worst-case.

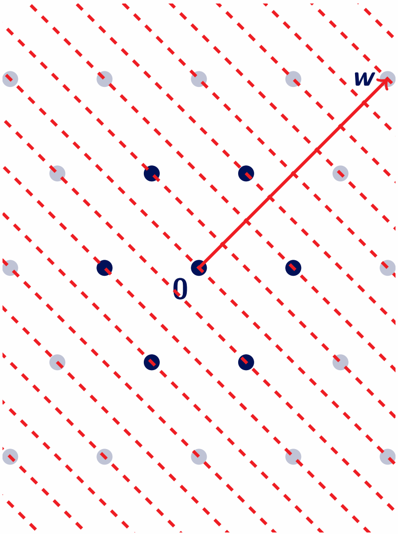



- Dual lattice:
  $$\mathcal{L}^* := \{\boldsymbol{w} \in \mathbb{R}^n : \forall \boldsymbol{v} \in \mathcal{L}, \langle \boldsymbol{w}, \boldsymbol{v} \rangle \in \mathbb{Z}\}$$




- Dual lattice:
  $$\mathcal{L}^* := \{w \in \mathbb{R}^n : \forall v \in \mathcal{L}, \langle w, v\rangle \in \mathbb{Z}\}$$

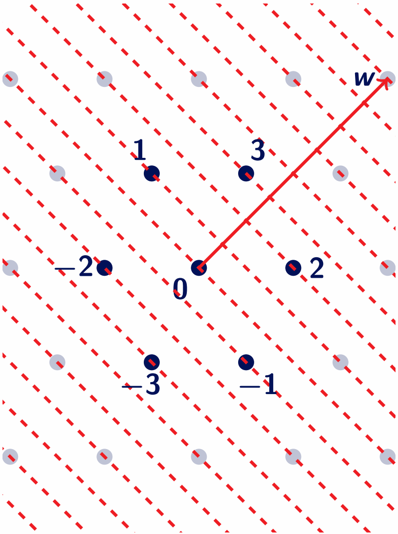
# Haviv-Regev (2014)




- Dual lattice:

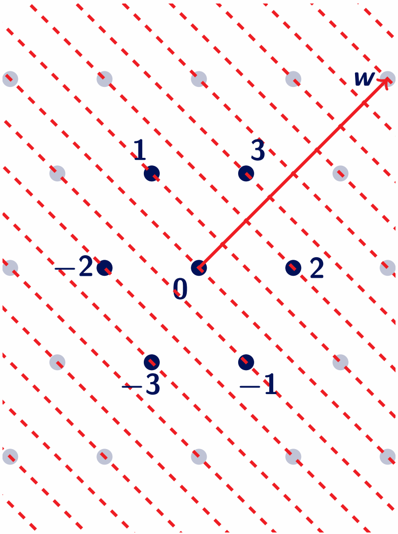
  $$\mathcal{L}^* := \{\boldsymbol{w} \in \mathbb{R}^n : \forall \boldsymbol{v} \in \mathcal{L}, \langle \boldsymbol{w}, \boldsymbol{v} \rangle \in \mathbb{Z}\}$$

- Idea: pick $\boldsymbol{w}_i \in \mathcal{L}_i^*$ that canonically orders $\mathcal{V}(\mathcal{L}_i)$ by values $\langle \boldsymbol{v}, \boldsymbol{w}_i \rangle$.




- Dual lattice:
  $$\mathcal{L}^* := \{\boldsymbol{w} \in \mathbb{R}^n : \forall \boldsymbol{v} \in \mathcal{L}, \langle \boldsymbol{w}, \boldsymbol{v} \rangle \in \mathbb{Z}\}$$
- Idea: pick $\boldsymbol{w}_i \in \mathcal{L}_i^*$ that canonically orders $\mathcal{V}(\mathcal{L}_i)$ by values $\langle \boldsymbol{v}, \boldsymbol{w}_i \rangle$.

# Haviv-Regev (2014)

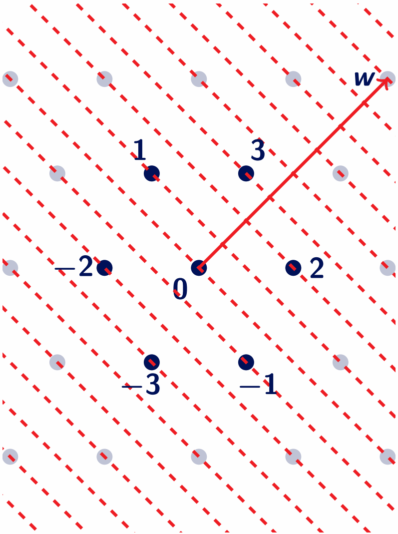



- Dual lattice:

  $$\mathcal{L}^* := \{\boldsymbol{w} \in \mathbb{R}^n : \forall \boldsymbol{v} \in \mathcal{L}, \langle \boldsymbol{w}, \boldsymbol{v} \rangle \in \mathbb{Z}\}$$

- Idea: pick $\boldsymbol{w_i} \in \mathcal{L}_i^*$ that canonically orders $\mathcal{V}(\mathcal{L}_i)$ by values $\langle \boldsymbol{v}, \boldsymbol{w_i} \rangle$.
- If $\boldsymbol{w}_2 = \boldsymbol{O}\boldsymbol{w}_1$, then $\mathcal{V}(\mathcal{L}_2) = \boldsymbol{O} \cdot \mathcal{V}(\mathcal{L}_1)$ (as ordered lists) $\implies$ recover $\boldsymbol{O}$.

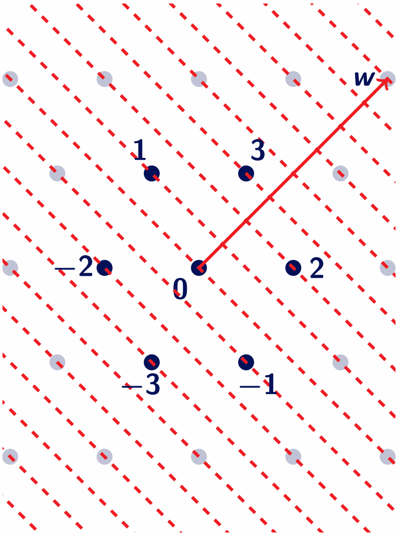
# Haviv-Regev (2014)




- Dual lattice:

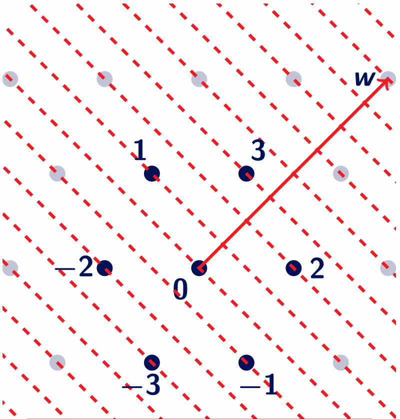
  $$\mathcal{L}^* := \{\boldsymbol{w} \in \mathbb{R}^n : \forall \boldsymbol{v} \in \mathcal{L}, \langle \boldsymbol{w}, \boldsymbol{v} \rangle \in \mathbb{Z}\}$$

- Idea: pick $\boldsymbol{w}_i \in \mathcal{L}_i^*$ that canonically orders $\mathcal{V}(\mathcal{L}_i)$ by values $\langle \boldsymbol{v}, \boldsymbol{w}_i \rangle$.
- If $\boldsymbol{w}_2 = \boldsymbol{O}\boldsymbol{w}_1$, then $\mathcal{V}(\mathcal{L}_2) = \boldsymbol{O} \cdot \mathcal{V}(\mathcal{L}_1)$ (as ordered lists) $\implies$ recover $\boldsymbol{O}$.
- Isolation Lemma: such a $\boldsymbol{w}_i \in \mathcal{L}_i^*$ exists among the $n^{O(n)}$ shortest vectors of $\mathcal{L}_i^*$.




- Dual lattice:
  $$\mathcal{L}^* := \{w \in \mathbb{R}^n : \forall v \in \mathcal{L}, \langle w, v\rangle \in \mathbb{Z}\}$$
- Idea: pick $w_i \in \mathcal{L}_i^*$ that canonically orders $\mathcal{V}(\mathcal{L}_i)$ by values $\langle v, w_i\rangle$.
- If $w_2 = Ow_1$, then $\mathcal{V}(\mathcal{L}_2) = O \cdot \mathcal{V}(\mathcal{L}_1)$ (as ordered lists) $\implies$ recover $O$.
- Isolation Lemma: such a $w_i \in \mathcal{L}_i^*$ exists among the $n^{O(n)}$ shortest vectors of $\mathcal{L}_i^*$.
- Haviv-Regev algorithm (informal):
  1. Compute $\mathcal{V}(\mathcal{L}_i)$ and $n^{O(n)}$ shortest vecs $S_i \subset \mathcal{L}_i^*$
  2. Isolate $w_1 \in S_1$, $w_2^{(1)}, \ldots, w_2^{(N)} \in S_2$.
  3. Recover isometries from $w_2^{(i)} = Ow_1$.

# Haviv-Regev (2014)




Time complexity: $n^{O(n)}$

OQ: Can we do better? $(2^{O(n)})$

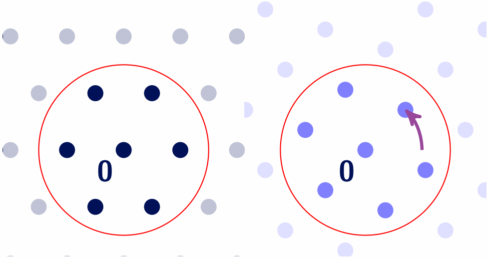
- Dual lattice:

$$\mathcal{L}^* := \{w \in \mathbb{R}^n : \forall v \in \mathcal{L}, \langle w, v \rangle \in \mathbb{Z}\}$$

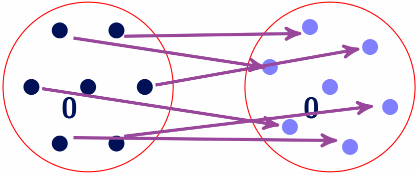
- Idea: pick $w_i \in \mathcal{L}_i^*$ that canonically orders $\mathcal{V}(\mathcal{L}_i)$ by values $\langle v, w_i \rangle$.
- If $w_2 = Ow_1$, then $\mathcal{V}(\mathcal{L}_2) = O \cdot \mathcal{V}(\mathcal{L}_1)$ (as ordered lists) $\implies$ recover $O$.
- Isolation Lemma: such a $w_i \in \mathcal{L}_i^*$ exists among the $n^{O(n)}$ shortest vectors of $\mathcal{L}_i^*$.
- Haviv-Regev algorithm (informal):
  1. Compute $\mathcal{V}(\mathcal{L}_i)$ and $n^{O(n)}$ shortest vecs $S_i \subset \mathcal{L}_i^*$
  2. Isolate $w_1 \in S_1$, $w_2^{(1)}, \ldots, w_2^{(N)} \in S_2$.
  3. Recover isometries from $w_2^{(i)} = Ow_1$.

# Open Questions

**Step 1:**
compute short vectors



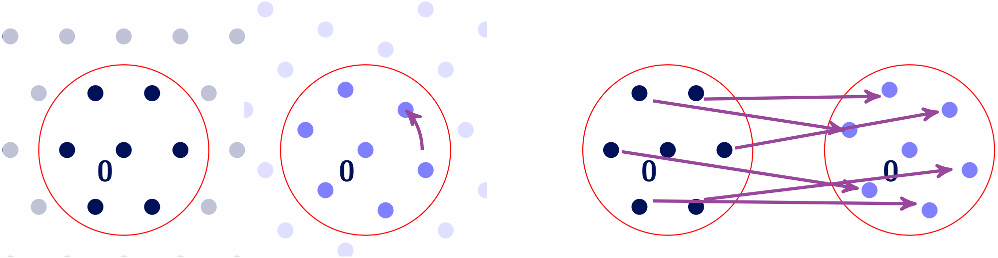

**Step 2:**
compute isometries between them



# Open Questions

Step 1:
compute short vectors

Step 2:
compute isometries between them

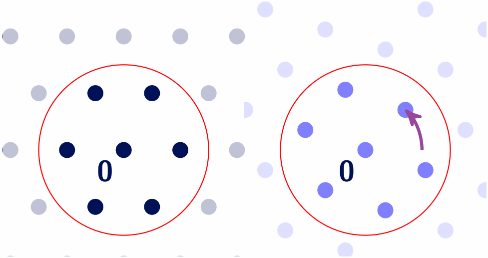



**Faster isometry finding:**

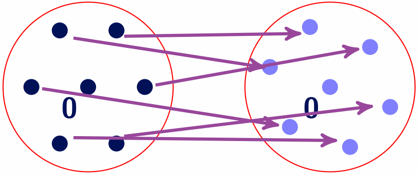
- Can we do step 2 in $2^{O(n)}$ time if searching for a single isometry?

# Open Questions

Step 1:
compute short vectors




Step 2:
compute isometries between them




**Faster isometry finding:**

- Can we do step 2 in $2^{O(n)}$ time if searching for a single isometry?

**Alternative approach?:**

- Can we solve LIP without first finding short vectors?

# Invariants for LIP

**Definition: search LIP (sLIP)**

Given two isomorphic lattices $\mathcal{L}_1, \mathcal{L}_2$, recover an orthonormal transformation $\boldsymbol{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\boldsymbol{O} \cdot \mathcal{L}_1 = \mathcal{L}_2$.

# LIP Variants

**Definition: search LIP (sLIP)**

Given two isomorphic lattices $\mathcal{L}_1, \mathcal{L}_2$, recover an orthonormal transformation $\boldsymbol{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\boldsymbol{O} \cdot \mathcal{L}_1 = \mathcal{L}_2$.

**Definition: decisional LIP (dLIP)**

Given two lattices $\mathcal{L}_1, \mathcal{L}_2$, determine whether $\mathcal{L}_1 \cong \mathcal{L}_2$ or not.

# LIP Variants

**Definition: search LIP (sLIP)**

Given two isomorphic lattices $\mathcal{L}_1, \mathcal{L}_2$, recover an orthonormal transformation $\boldsymbol{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\boldsymbol{O} \cdot \mathcal{L}_1 = \mathcal{L}_2$.

**Definition: decisional LIP (dLIP)**

Given two lattices $\mathcal{L}_1, \mathcal{L}_2$, determine whether $\mathcal{L}_1 \cong \mathcal{L}_2$ or not.

**Definition: distinguish LIP ($\Delta$LIP)**

Let $\mathcal{L}_1, \mathcal{L}_2$ be two non-isomorphic lattices and let $b \leftarrow \{1, 2\}$ uniform. Given $\mathcal{L} \in [\mathcal{L}_b]$, recover $b$.

# LIP Variants

**Definition: search LIP (sLIP)**

Given two isomorphic lattices $\mathcal{L}_1, \mathcal{L}_2$, recover an orthonormal transformation $O \in \mathcal{O}_n(\mathbb{R})$ such that $O \cdot \mathcal{L}_1 = \mathcal{L}_2$.

**Definition: decisional LIP (dLIP)**

Given two lattices $\mathcal{L}_1, \mathcal{L}_2$, determine whether $\mathcal{L}_1 \cong \mathcal{L}_2$ or not.

**Definition: distinguish LIP ($\Delta$LIP)**

Let $\mathcal{L}_1, \mathcal{L}_2$ be two non-isomorphic lattices and let $b \leftarrow \{1, 2\}$ uniform. Given $\mathcal{L} \in [\mathcal{L}_b]$, recover $b$.

- Distinguishing variant is useful for security proofs: one can replace $[\mathcal{L}_1]$ by $[\mathcal{L}_2]$ in security game.

# Invariants

- Disclaimer: we only consider integral lattices ($\boldsymbol{B}^\top \boldsymbol{B} \in \mathbb{Z}^{\boldsymbol{n}\times\boldsymbol{n}}$)

# Invariants

- Disclaimer: we only consider integral lattices ($\boldsymbol{B}^\top \boldsymbol{B} \in \mathbb{Z}^{n \times n}$)

Arithmetic Invariants ($\mathbf{ari}(\mathcal{L})$)

- $\mathbf{det}(\mathcal{L}) = \mathbf{det}(\mathcal{L}_b)$.
- $\mathbf{gcd}(\mathcal{L}) := \mathbf{gcd}\{\langle x, y\rangle : x, y \in \mathcal{L}\}$
- parity $\mathbf{par}(\mathcal{L}) = \mathbf{gcd}\{\|x\|^2 : x \in \mathcal{L}\} / \mathbf{gcd}(\mathcal{L})$

- Disclaimer: we only consider integral lattices ($B^\top B \in \mathbb{Z}^{n\times n}$)

Arithmetic Invariants ($\mathbf{ari}(\mathcal{L})$)

- $\det(\mathcal{L}) = \det(\mathcal{L}_b)$.
- $\gcd(\mathcal{L}) := \gcd\{\langle x, y\rangle : x, y \in \mathcal{L}\}$
- parity $\mathrm{par}(\mathcal{L}) = \gcd\{\|x\|^2 : x \in \mathcal{L}\} / \gcd(\mathcal{L})$
- Equivalence over $R \supset \mathbb{Z}$, $U \in \mathrm{GL}_n(R)$, $R \in \{\mathbb{R}, \mathbb{Q}, \forall p\ \mathbb{Q}_p, \underbrace{\forall p\ \mathbb{Z}_p}_{\text{Genus}}\}$

- **Disclaimer:** we only consider integral lattices ($\boldsymbol{B}^\top \boldsymbol{B} \in \mathbb{Z}^{n\times n}$)

Arithmetic Invariants ($\mathbf{ari}(\mathcal{L})$)

- $\det(\mathcal{L}) = \det(\mathcal{L}_b)$.
- $\gcd(\mathcal{L}) := \gcd\{\langle x, y\rangle : x, y \in \mathcal{L}\}$
- parity $\mathrm{par}(\mathcal{L}) = \gcd\{\|x\|^2 : x \in \mathcal{L}\}/\gcd(\mathcal{L})$
- Equivalence over $R \supset \mathbb{Z}$, $U \in \mathrm{GL}_n(R)$, $R \in \{\mathbb{R}, \mathbb{Q}, \forall p\ \mathbb{Q}_p, \underbrace{\forall p\ \mathbb{Z}_p}_{\text{Genus}}\}$

**Lemma:**

If $\mathbf{ari}(\mathcal{L}_1) \neq \mathbf{ari}(\mathcal{L}_2)$, then ***dLIP*** and **Δ**LIP with $\mathcal{L}_1, \mathcal{L}_2$ can be solved efficiently.

- Disclaimer: we only consider integral lattices ($B^\top B \in \mathbb{Z}^{n\times n}$)

Arithmetic Invariants ($\mathbf{ari}(\mathcal{L})$)

- $\det(\mathcal{L}) = \det(\mathcal{L}_b)$.
- $\gcd(\mathcal{L}) := \gcd\{\langle x, y\rangle : x, y \in \mathcal{L}\}$
- parity $\mathrm{par}(\mathcal{L}) = \gcd\{\|x\|^2 : x \in \mathcal{L}\} / \gcd(\mathcal{L})$
- Equivalence over $R \supset \mathbb{Z}$, $U \in \mathrm{GL}_n(R)$, $R \in \{\mathbb{R}, \mathbb{Q}, \forall p\ \mathbb{Q}_p, \underbrace{\forall p\ \mathbb{Z}_p}_{\text{Genus}}\}$

**Lemma:**

If $\mathbf{ari}(\mathcal{L}_1) \neq \mathbf{ari}(\mathcal{L}_2)$, then ***dLIP*** and **Δ**LIP with $\mathcal{L}_1, \mathcal{L}_2$ can be solved efficiently.

$\Rightarrow$ lattices must have same (efficiently computable) invariants

**$p$-adic integers:**

For a prime $p$ the $p$-adic integers $\mathbb{Z}_p$ are given by formal series, i.e.,

$$\mathbb{Z}_p = \left\{ \sum_{i=0}^{\infty} a_i p^i, \quad \text{with } 0 \leq a_i < p \right\}$$

**$p$-adic integers:**

For a prime $p$ the $p$-adic integers $\mathbb{Z}_p$ are given by formal series, i.e.,

$$\mathbb{Z}_p = \left\{ \sum_{i=0}^{\infty} a_i p^i, \quad \text{with } 0 \leq a_i < p \right\}$$

**Genus:**

The genus $\mathbf{gen}(\mathcal{L})$ of a lattice $\mathcal{L}$ consists of all lattices that are equivalent over $\mathbb{R}$ and over $\mathbb{Z}_p$ for all primes $p$

# Genus

**$p$-adic integers:**

For a prime $p$ the $p$-adic integers $\mathbb{Z}_p$ are given by formal series, i.e.,

$$\mathbb{Z}_p = \left\{ \sum_{i=0}^{\infty} a_i p^i, \quad \text{with } 0 \leq a_i < p \right\}$$

**Genus:**

The genus $\mathbf{gen}(\mathcal{L})$ of a lattice $\mathcal{L}$ consists of all lattices that are equivalent over $\mathbb{R}$ and over $\mathbb{Z}_p$ for all primes $p$

- Equivalent over $\mathbb{R}$ $\Leftrightarrow$ same rank

# Genus

**$p$-adic integers:**

For a prime $p$ the $p$-adic integers $\mathbb{Z}_p$ are given by formal series, i.e.,

$$\mathbb{Z}_p = \left\{ \sum_{i=0}^{\infty} a_i p^i, \quad \text{with } 0 \leq a_i < p \right\}$$

**Genus:**

The genus $\mathrm{gen}(\mathcal{L})$ of a lattice $\mathcal{L}$ consists of all lattices that are equivalent over $\mathbb{R}$ and over $\mathbb{Z}_p$ for all primes $p$

- Equivalent over $\mathbb{R}$ $\Leftrightarrow$ same rank
- Equivalent over $\mathbb{Z}_p$ $\Leftrightarrow \mathbb{Z}_p \otimes \mathcal{L}_1 \cong \mathbb{Z}_p \otimes \mathcal{L}_2$

$$\Leftrightarrow U^\top G_1 U = G_2 \text{ for } U \in \mathcal{GL}_n(\mathbb{Z}_p).$$

**$p$-adic integers:**

For a prime $p$ the $p$-adic integers $\mathbb{Z}_p$ are given by formal series, i.e.,

$$\mathbb{Z}_p = \left\{ \sum_{i=0}^{\infty} a_i p^i, \quad \text{with } 0 \le a_i < p \right\}$$

**Genus:**

The genus $\mathbf{gen}(\mathcal{L})$ of a lattice $\mathcal{L}$ consists of all lattices that are equivalent over $\mathbb{R}$ and over $\mathbb{Z}_p$ for all primes $p$

- Equivalent over $\mathbb{R}$ $\Leftrightarrow$ same rank
- Equivalent over $\mathbb{Z}_p$ $\Leftrightarrow$ $\mathbb{Z}_p \otimes \mathcal{L}_1 \cong \mathbb{Z}_p \otimes \mathcal{L}_2$

  $$\Leftrightarrow U^\top G_1 U = G_2 \text{ for } U \in \mathcal{GL}_n(\mathbb{Z}_p).$$
- Covers all the other known arithmetic invariants*

**$p$-adic integers:**

For a prime $p$ the $p$-adic integers $\mathbb{Z}_p$ are given by formal series, i.e.,

$$\mathbb{Z}_p = \left\{ \sum_{i=0}^{\infty} a_i p^i, \quad \text{with } 0 \leq a_i < p \right\}$$

**Genus:**

The genus $\mathbf{gen}(\mathcal{L})$ of a lattice $\mathcal{L}$ consists of all lattices that are equivalent over $\mathbb{R}$ and over $\mathbb{Z}_p$ for all primes $p$

- Equivalent over $\mathbb{R}$ $\Leftrightarrow$ same rank
- Equivalent over $\mathbb{Z}_p$ $\Leftrightarrow \mathbb{Z}_p \otimes \mathcal{L}_1 \cong \mathbb{Z}_p \otimes \mathcal{L}_2$

$$\Leftrightarrow U^\top G_1 U = G_2 \text{ for } U \in \mathcal{GL}_n(\mathbb{Z}_p).$$

- Covers all the other known arithmetic invariants*

* (we assume here the genus does not split into multiple spinor genera)

# How to compute genus equivalence?

- We consider $\boldsymbol{p} \geq \mathbf{3}$.

# How to compute genus equivalence?

- We consider $p \geq 3$.
- Idea: over $\mathbb{Z}_p$ the gram matrix is efficiently diagonalizable.

$$G \cong_{\mathbb{Z}_p} G_1 \oplus pG_p \oplus p^2 G_p \oplus \ldots \oplus p^k G_{p^k},$$

where $\det(G_q) \neq 0 \bmod p$, and each $G_q$ is a diagonal matrix.

## How to compute genus equivalence?

- We consider $p \geq 3$.
- Idea: over $\mathbb{Z}_p$ the gram matrix is efficiently diagonalizable.

$$G \cong_{\mathbb{Z}_p} G_1 \oplus pG_p \oplus p^2 G_p \oplus \ldots \oplus p^k G_{p^k},$$

  where $\det(G_q) \neq 0 \bmod p$, and each $G_q$ is a diagonal matrix.
- For the diagonal matrices $G_q$, $\mathbb{Z}_p$ equivalence is fully determined by $\dim(G_q)$ and the Legendre symbol $\left(\frac{\det(G_q)}{p}\right)$.

# How to compute genus equivalence?

- We consider $p \geq 3$.
- Idea: over $\mathbb{Z}_p$ the gram matrix is efficiently diagonalizable.

$$G \cong_{\mathbb{Z}_p} G_1 \oplus pG_p \oplus p^2 G_p \oplus \ldots \oplus p^k G_{p^k},$$

  where $\det(G_q) \neq 0 \bmod p$, and each $G_q$ is a diagonal matrix.
- For the diagonal matrices $G_q$, $\mathbb{Z}_p$ equivalence is fully determined by $\dim(G_q)$ and the Legendre symbol $\left(\frac{\det(G_q)}{p}\right)$.
- $G \cong_{\mathbb{Z}_p} G'$ if the above values match for all $q = p^i$.

# How to compute genus equivalence?

- We consider $p \geq 3$.
- Idea: over $\mathbb{Z}_p$ the gram matrix is efficiently diagonalizable.

$$G \cong_{\mathbb{Z}_p} G_1 \oplus pG_p \oplus p^2 G_p \oplus \ldots \oplus p^k G_{p^k},$$

  where $\det(G_q) \neq 0 \bmod p$, and each $G_q$ is a diagonal matrix.
- For the diagonal matrices $G_q$, $\mathbb{Z}_p$ equivalence is fully determined by $\dim(G_q)$ and the Legendre symbol $\left(\frac{\det(G_q)}{p}\right)$.
- $G \cong_{\mathbb{Z}_p} G'$ if the above values match for all $q = p^i$.
- For $p \nmid \det(G)$ we have $\dim(G_1) = \dim(G)$ and $\left(\frac{\det(G_1)}{p}\right) = \left(\frac{\det(G)}{p}\right)$.

# How to compute genus equivalence?

- We consider $p \geq 3$.
- Idea: over $\mathbb{Z}_p$ the gram matrix is efficiently diagonalizable.

$$G \cong_{\mathbb{Z}_p} G_1 \oplus pG_p \oplus p^2 G_p \oplus \ldots \oplus p^k G_{p^k},$$

  where $\det(G_q) \neq 0 \bmod p$, and each $G_q$ is a diagonal matrix.
- For the diagonal matrices $G_q$, $\mathbb{Z}_p$ equivalence is fully determined by $\dim(G_q)$ and the Legendre symbol $\left(\frac{\det(G_q)}{p}\right)$.
- $G \cong_{\mathbb{Z}_p} G'$ if the above values match for all $q = p^i$.
- For $p \nmid \det(G)$ we have $\dim(G_1) = \dim(G)$ and $\left(\frac{\det(G_1)}{p}\right) = \left(\frac{\det(G)}{p}\right)$.
- So only have to consider $p \mid \det(G)$ (needs factorization)

# How to compute genus equivalence?

- We consider $p \geq 3$.
- Idea: over $\mathbb{Z}_p$ the gram matrix is efficiently diagonalizable.

$$G \cong_{\mathbb{Z}_p} G_1 \oplus pG_p \oplus p^2 G_p \oplus \ldots \oplus p^k G_{p^k},$$

  where $\det(G_q) \neq 0 \bmod p$, and each $G_q$ is a diagonal matrix.
- For the diagonal matrices $G_q$, $\mathbb{Z}_p$ equivalence is fully determined by $\dim(G_q)$ and the Legendre symbol $\left(\frac{\det(G_q)}{p}\right)$.
- $G \cong_{\mathbb{Z}_p} G'$ if the above values match for all $q = p^i$.
- For $p \nmid \det(G)$ we have $\dim(G_1) = \dim(G)$ and $\left(\frac{\det(G_1)}{p}\right) = \left(\frac{\det(G)}{p}\right)$.
- So only have to consider $p \mid \det(G)$ (needs factorization)
- For $p = 2$ block diagonalizable and a few additional rules.

# How to compute genus equivalence?

- We consider $p \geq 3$.
- Idea: over $\mathbb{Z}_p$ the gram matrix is efficiently diagonalizable.

$$G \cong_{\mathbb{Z}_p} G_1 \oplus pG_p \oplus p^2 G_p \oplus \ldots \oplus p^k G_{p^k},$$

  where $\det(G_q) \neq 0 \bmod p$, and each $G_q$ is a diagonal matrix.
- For the diagonal matrices $G_q$, $\mathbb{Z}_p$ equivalence is fully determined by $\dim(G_q)$ and the Legendre symbol $\left(\frac{\det(G_q)}{p}\right)$.
- $G \cong_{\mathbb{Z}_p} G'$ if the above values match for all $q = p^i$.
- For $p \nmid \det(G)$ we have $\dim(G_1) = \dim(G)$ and $\left(\frac{\det(G_1)}{p}\right) = \left(\frac{\det(G)}{p}\right)$.
- So only have to consider $p \mid \det(G)$ (needs factorization)
- For $p = 2$ block diagonalizable and a few additional rules.
- How restricting is the genus invariant?

# Mass formula and the size of a genus

**Theorem:** Smith-Minkowski-Siegel mass formula (Siegel, 1935)

Any genus $\mathcal{G}$ contains a finite number of isom. classes and its mass

$$M(\mathcal{G}) := \sum_{[\mathcal{L}]\in\mathcal{G}} \frac{1}{|\mathrm{Aut}(\mathcal{L})|},$$

is efficiently computable given the prime factorization of $\det(\mathcal{G})^2$.

# Mass formula and the size of a genus

**Theorem:** Smith-Minkowski-Siegel mass formula (Siegel, 1935)

Any genus $\mathcal{G}$ contains a finite number of isom. classes and its mass

$$\boldsymbol{M}(\mathcal{G}) := \sum_{[\mathcal{L}]\in\mathcal{G}} \frac{1}{|\mathrm{Aut}(\mathcal{L})|},$$

is efficiently computable given the prime factorization of $\mathbf{det}(\mathcal{G})^2$.

- **Lemma:** $|\mathcal{G}| \geq 2M(\mathcal{G})$. Proof: $|\mathrm{Aut}(\mathcal{L})| \geq 2$. □

# Mass formula and the size of a genus

**Theorem:** Smith-Minkowski-Siegel mass formula (Siegel, 1935)

Any genus $\mathcal{G}$ contains a finite number of isom. classes and its mass

$$M(\mathcal{G}) := \sum_{[\mathcal{L}]\in\mathcal{G}} \frac{1}{|\mathrm{Aut}(\mathcal{L})|},$$

is efficiently computable given the prime factorization of $\det(\mathcal{G})^2$.

- Lemma: $|\mathcal{G}| \geq 2M(\mathcal{G})$. Proof: $|\mathrm{Aut}(\mathcal{L})| \geq 2$. □
- Example: $M(\mathrm{Gen}(\mathbb{Z}^{32})) \approx 4.33 \cdot 10^{16}$

  $M(\mathrm{Gen}(\mathbb{Z}^{40})) \approx 1.21 \cdot 10^{63}$

# Mass formula and the size of a genus

**Theorem:** Smith-Minkowski-Siegel mass formula (Siegel, 1935)

Any genus $\mathcal{G}$ contains a finite number of isom. classes and its mass

$$M(\mathcal{G}) := \sum_{[\mathcal{L}] \in \mathcal{G}} \frac{1}{|\mathrm{Aut}(\mathcal{L})|},$$

is efficiently computable given the prime factorization of $\det(\mathcal{G})^2$.

- Lemma: $|\mathcal{G}| \geq 2M(\mathcal{G})$. Proof: $|\mathrm{Aut}(\mathcal{L})| \geq 2$. □
- Example: $M(\mathrm{Gen}(\mathbb{Z}^{32})) \approx 4.33 \cdot 10^{16}$

  $M(\mathrm{Gen}(\mathbb{Z}^{40})) \approx 1.21 \cdot 10^{63}$
- Grows fast: $M(\mathcal{G}) \geq n^{\Omega(n^2)}$ as $n \to \infty$

# Mass formula and the size of a genus

**Theorem:** Smith-Minkowski-Siegel mass formula (Siegel, 1935)

Any genus $\mathcal{G}$ contains a finite number of isom. classes and its mass

$$M(\mathcal{G}) := \sum_{[\mathcal{L}]\in\mathcal{G}} \frac{1}{|\mathrm{Aut}(\mathcal{L})|},$$

is efficiently computable given the prime factorization of $\det(\mathcal{G})^2$.

- Lemma: $|\mathcal{G}| \geq 2M(\mathcal{G})$. Proof: $|\mathrm{Aut}(\mathcal{L})| \geq 2$. □
- Example: $M(\mathrm{Gen}(\mathbb{Z}^{32})) \approx 4.33 \cdot 10^{16}$
  $M(\mathrm{Gen}(\mathbb{Z}^{40})) \approx 1.21 \cdot 10^{63}$
- Grows fast: $M(\mathcal{G}) \geq n^{\Omega(n^2)}$ as $n \to \infty$
- Enormous number of isomorphism classes in same genus

# Mass formula and the size of a genus

**Theorem:** Smith-Minkowski-Siegel mass formula (Siegel, 1935)

Any genus $\mathcal{G}$ contains a finite number of isom. classes and its mass

$$M(\mathcal{G}) := \sum_{[\mathcal{L}] \in \mathcal{G}} \frac{1}{|\mathrm{Aut}(\mathcal{L})|},$$

is efficiently computable given the prime factorization of $\det(\mathcal{G})^2$.

- Lemma: $|\mathcal{G}| \geq 2M(\mathcal{G})$. Proof: $|\mathrm{Aut}(\mathcal{L})| \geq 2$. □
- Example: $M(\mathrm{Gen}(\mathbb{Z}^{32})) \approx 4.33 \cdot 10^{16}$

  $M(\mathrm{Gen}(\mathbb{Z}^{40})) \approx 1.21 \cdot 10^{63}$
- Grows fast: $M(\mathcal{G}) \geq n^{\Omega(n^2)}$ as $n \to \infty$
- Enormous number of isomorphism classes in same genus
- Question: do these behave like random lattices?

# Random distribution over genus

**Definition: distribution over Genus**

Let $w(\mathcal{L}) =: 1/|\mathrm{Aut}(\mathcal{L})|$. For a genus $\mathcal{G}$ let $\mathcal{D}(\mathcal{G})$ be the distribution such that each class $[\mathcal{L}] \in \mathcal{G}$ is sampled with probability $\frac{w(\mathcal{L})}{M(\mathcal{G})}$.

# Random distribution over genus

**Definition: distribution over Genus**

Let $w(\mathcal{L}) =: 1/|\mathrm{Aut}(\mathcal{L})|$. For a genus $\mathcal{G}$ let $\mathcal{D}(\mathcal{G})$ be the distribution such that each class $[\mathcal{L}] \in \mathcal{G}$ is sampled with probability $\frac{w(\mathcal{L})}{M(\mathcal{G})}$.

- Coincides with the distribution of random lattices (Haar measure) restricted to a single genus.

# Random distribution over genus

**Definition: distribution over Genus**

Let $w(\mathcal{L}) =: 1/|\mathrm{Aut}(\mathcal{L})|$. For a genus $\mathcal{G}$ let $\mathcal{D}(\mathcal{G})$ be the distribution such that each class $[\mathcal{L}] \in \mathcal{G}$ is sampled with probability $\frac{w(\mathcal{L})}{M(\mathcal{G})}$.

- Coincides with the distribution of random lattices (Haar measure) restricted to a single genus.

**Theorem (informal): good geometric properties [vW, soon on eprint]**

For any genus $\mathcal{G}$ (satisfying some minor properties), samples from $\mathcal{D}(\mathcal{G})$ have a packing density, covering radius and smoothing parameter similar to that of random lattices.

# Random distribution over genus

**Definition:** distribution over Genus

Let $w(\mathcal{L}) =: 1/|\mathrm{Aut}(\mathcal{L})|$. For a genus $\mathcal{G}$ let $\mathcal{D}(\mathcal{G})$ be the distribution such that each class $[\mathcal{L}] \in \mathcal{G}$ is sampled with probability $\frac{w(\mathcal{L})}{M(\mathcal{G})}$.

- Coincides with the distribution of random lattices (Haar measure) restricted to a single genus.

**Theorem (informal):** good geometric properties [vW, soon on eprint]

For any genus $\mathcal{G}$ (satisfying some minor properties), samples from $\mathcal{D}(\mathcal{G})$ have a packing density, covering radius and smoothing parameter similar to that of random lattices.

- Proven via other Mass formulas by Siegel (1935)

# Random distribution over genus

**Definition:** distribution over Genus

Let $w(\mathcal{L}) =: 1/|\mathrm{Aut}(\mathcal{L})|$. For a genus $\mathcal{G}$ let $\mathcal{D}(\mathcal{G})$ be the distribution such that each class $[\mathcal{L}] \in \mathcal{G}$ is sampled with probability $\frac{w(\mathcal{L})}{M(\mathcal{G})}$.

- Coincides with the distribution of random lattices (Haar measure) restricted to a single genus.

**Theorem (informal):** good geometric properties [vW, soon on eprint]

For any genus $\mathcal{G}$ (satisfying some minor properties), samples from $\mathcal{D}(\mathcal{G})$ have a packing density, covering radius and smoothing parameter similar to that of random lattices.

- Proven via other Mass formulas by Siegel (1935)
- Heuristically, these are the hardest lattices to distinguish.

# Kneser $p$-neighbouring (1957) and sampling

- Two integral lattices $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours $\mathcal{L}_1 \sim_p \mathcal{L}_2$ if

$$[\mathcal{L}_1 : \mathcal{L}_1 \cap \mathcal{L}_2] = [\mathcal{L}_2 : \mathcal{L}_1 \cap \mathcal{L}_2] = p.$$

# Kneser $p$-neighbouring (1957) and sampling

- Two integral lattices $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours $\mathcal{L}_1 \sim_p \mathcal{L}_2$ if

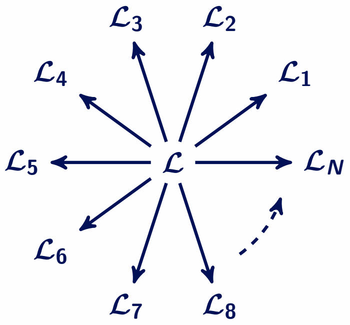
$$[\mathcal{L}_1 : \mathcal{L}_1 \cap \mathcal{L}_2] = [\mathcal{L}_2 : \mathcal{L}_1 \cap \mathcal{L}_2] = p.$$

- If $\mathcal{L}_1 \sim_p \mathcal{L}_2$ then $\mathbf{Gen}(\mathcal{L}_1) = \mathbf{Gen}(\mathcal{L}_2)$.

- Two integral lattices $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours $\mathcal{L}_1 \sim_p \mathcal{L}_2$ if

$$[\mathcal{L}_1 : \mathcal{L}_1 \cap \mathcal{L}_2] = [\mathcal{L}_2 : \mathcal{L}_1 \cap \mathcal{L}_2] = p.$$

- If $\mathcal{L}_1 \sim_p \mathcal{L}_2$ then $\mathbf{Gen}(\mathcal{L}_1) = \mathbf{Gen}(\mathcal{L}_2)$.
- A lattice has $\sim p^{n-2}$ $p$-neighours ($\leftrightarrow$ isotropic lines in $\mathcal{L}/p\mathcal{L}$).

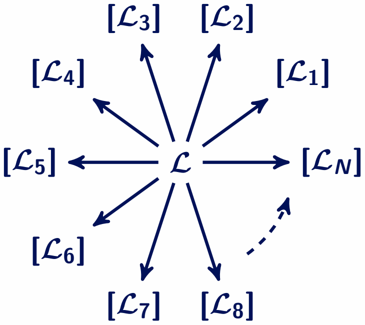


# Kneser $p$-neighbouring (1957) and sampling

- Two integral lattices $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours $\mathcal{L}_1 \sim_p \mathcal{L}_2$ if

$$[\mathcal{L}_1 : \mathcal{L}_1 \cap \mathcal{L}_2] = [\mathcal{L}_2 : \mathcal{L}_1 \cap \mathcal{L}_2] = p.$$

- If $\mathcal{L}_1 \sim_p \mathcal{L}_2$ then $\mathbf{Gen}(\mathcal{L}_1) = \mathbf{Gen}(\mathcal{L}_2)$.
- A lattice has $\sim p^{n-2}$ $p$-neighours ($\leftrightarrow$ isotropic lines in $\mathcal{L}/p\mathcal{L}$).

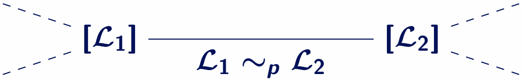


# Kneser $p$-neighbouring (1957) and sampling

- Two integral lattices $\mathcal{L}_1, \mathcal{L}_2$ are **$p$-neighbours** $\mathcal{L}_1 \sim_p \mathcal{L}_2$ if

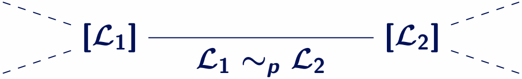
$$[\mathcal{L}_1 : \mathcal{L}_1 \cap \mathcal{L}_2] = [\mathcal{L}_2 : \mathcal{L}_1 \cap \mathcal{L}_2] = p.$$

- If $\mathcal{L}_1 \sim_p \mathcal{L}_2$ then $\mathbf{Gen}(\mathcal{L}_1) = \mathbf{Gen}(\mathcal{L}_2)$.
- A lattice has $\sim p^{n-2}$ $p$-neighours ($\leftrightarrow$ isotropic lines in $\mathcal{L}/p\mathcal{L}$).
- Turns any genus into a graph with nodes $[\mathcal{L}_1], \ldots, [\mathcal{L}_N]$ and an edge $([\mathcal{L}_i], [\mathcal{L}_j])$ if $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours up to isometry.



# Kneser $p$-neighbouring (1957) and sampling

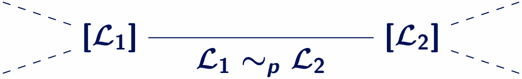
- Two integral lattices $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours $\mathcal{L}_1 \sim_p \mathcal{L}_2$ if
$$[\mathcal{L}_1 : \mathcal{L}_1 \cap \mathcal{L}_2] = [\mathcal{L}_2 : \mathcal{L}_1 \cap \mathcal{L}_2] = p.$$
- If $\mathcal{L}_1 \sim_p \mathcal{L}_2$ then $\mathbf{Gen}(\mathcal{L}_1) = \mathbf{Gen}(\mathcal{L}_2)$.
- A lattice has $\sim p^{n-2}$ $p$-neighours ($\leftrightarrow$ isotropic lines in $\mathcal{L}/p\mathcal{L}$).
- Turns any genus into a graph with nodes $[\mathcal{L}_1], \ldots, [\mathcal{L}_N]$ and an edge $([\mathcal{L}_i], [\mathcal{L}_j])$ if $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours up to isometry.




- Random walk: $\mathcal{L}_1 \sim_p \mathcal{L}_2 \sim_p \ldots \sim_p \mathcal{L}_k$ where $\mathcal{L}_{i+1}$ is a uniformly randomly $p$-neighbour of $\mathcal{L}_i$.

# Kneser $p$-neighbouring (1957) and sampling

- Two integral lattices $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours $\mathcal{L}_1 \sim_p \mathcal{L}_2$ if
$$[\mathcal{L}_1 : \mathcal{L}_1 \cap \mathcal{L}_2] = [\mathcal{L}_2 : \mathcal{L}_1 \cap \mathcal{L}_2] = p.$$
- If $\mathcal{L}_1 \sim_p \mathcal{L}_2$ then $\mathbf{Gen}(\mathcal{L}_1) = \mathbf{Gen}(\mathcal{L}_2)$.
- A lattice has $\sim p^{n-2}$ $p$-neighours ($\leftrightarrow$ isotropic lines in $\mathcal{L}/p\mathcal{L}$).
- Turns any genus into a graph with nodes $[\mathcal{L}_1], \ldots, [\mathcal{L}_N]$ and an edge $([\mathcal{L}_i], [\mathcal{L}_j])$ if $\mathcal{L}_1, \mathcal{L}_2$ are $p$-neighbours up to isometry.




- Random walk: $\mathcal{L}_1 \sim_p \mathcal{L}_2 \sim_p \ldots \sim_p \mathcal{L}_k$ where $\mathcal{L}_{i+1}$ is a uniformly randomly $p$-neighbour of $\mathcal{L}_i$.
- For large enough $p$, a random walk has limit distribution $\mathcal{D}(\mathcal{G})$.
  $\implies$ efficient sampling algorithm for $\mathcal{D}(\mathcal{G})$.

# Open Questions

**WC-AC reductions:**

- the random case $[\mathcal{L}] \leftarrow \mathcal{D}(\mathcal{G})$ is heuristically the hardest.
- from any class $[\mathcal{L}] \in \mathcal{G}$ we can efficiently step to a random class.

Can we make a worst-case to average-case reduction within a genus?

Example: SVP, SIVP, LIP

# Open Questions

WC-AC reductions:

- the random case $[\mathcal{L}] \leftarrow \mathcal{D}(\mathcal{G})$ is heuristically the hardest.
- from any class $[\mathcal{L}] \in \mathcal{G}$ we can efficiently step to a random class.

Can we make a worst-case to average-case reduction within a genus?

Example: SVP, SIVP, LIP

Better invariants:

- Can we construct stronger efficiently computable invariants?

# Open Questions

**WC-AC reductions:**

- the random case $[\mathcal{L}] \leftarrow \mathcal{D}(\mathcal{G})$ is heuristically the hardest.
- from any class $[\mathcal{L}] \in \mathcal{G}$ we can efficiently step to a random class.

Can we make a worst-case to average-case reduction within a genus?

Example: SVP, SIVP, LIP

**Better invariants:**

- Can we construct stronger efficiently computable invariants?

**Structured case:**

What about module lattices?

- Can we find (significantly) better algorithms for module-LIP?
- How strong is a 'module-genus' invariant?

# Recap

- LIP is well studied from a mathematical perspective (long ago!).

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors
  2. Find isometries between them

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors
  2. Find isometries between them
- The genus is the strongest* known efficient invariant for LIP

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors
  2. Find isometries between them
- The genus is the strongest* known efficient invariant for LIP
  - Is not too restricting on the geometry

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors
  2. Find isometries between them
- The genus is the strongest* known efficient invariant for LIP
  - Is not too restricting on the geometry
  - Has a deep theory behind it: randomness, $p$-neighbouring, mass formula's

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors
  2. Find isometries between them
- The genus is the strongest* known efficient invariant for LIP
  - Is not too restricting on the geometry
  - Has a deep theory behind it: randomness, $p$-neighbouring, mass formula's
  - Lots of open questions related to the genus

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors
  2. Find isometries between them
- The genus is the strongest* known efficient invariant for LIP
  - Is not too restricting on the geometry
  - Has a deep theory behind it: randomness, $p$-neighbouring, mass formula's
  - Lots of open questions related to the genus
- An exciting new area for mathematical cryptology!

# Recap

- LIP is well studied from a mathematical perspective (long ago!).
- Classical algorithms to solve LIP
  1. Compute short vectors
  2. Find isometries between them
- The genus is the strongest* known efficient invariant for LIP
  - Is not too restricting on the geometry
  - Has a deep theory behind it: randomness, $p$-neighbouring, mass formula's
  - Lots of open questions related to the genus
- An exciting new area for mathematical cryptology!

Thanks!