# Lattice-based cryptography,

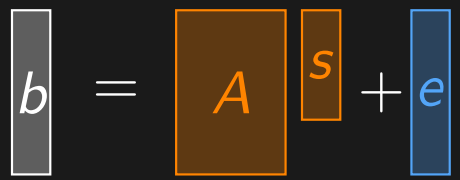
## the picture way

Alice Pellet-Mary

SCN 2024, Amalfi


# Two lattice worlds

## Matrix formalism

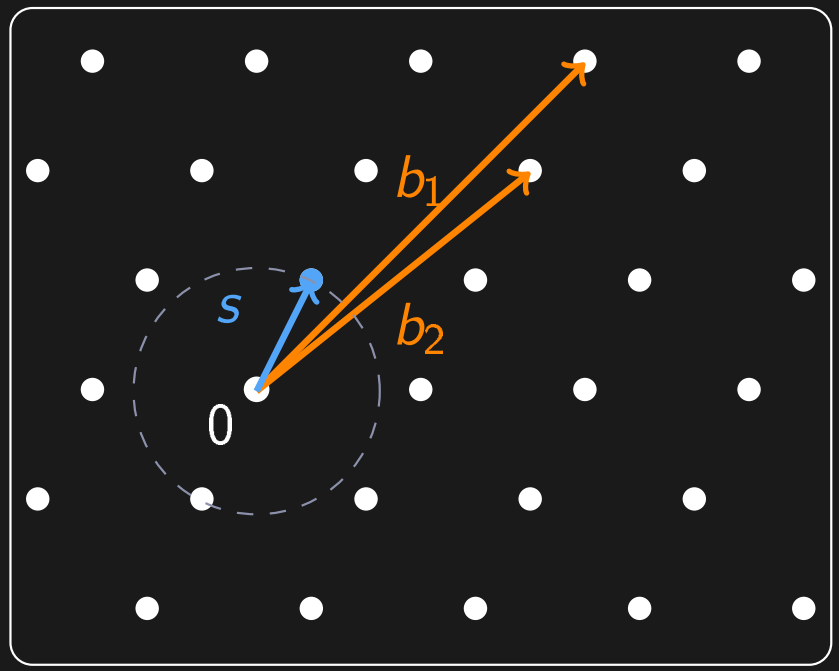



Orange: uniform in $\mathbb{Z}/q\mathbb{Z}$

Blue: uniform in $\{-1, 0, 1\}$

**LWE**: given $A$ and $b$, find $s$

## Geometric formalism




**SVP**: given $(b_1, b_2)$, find $s$

# Two lattice worlds

## Matrix formalism

$$b = A\,s + e$$

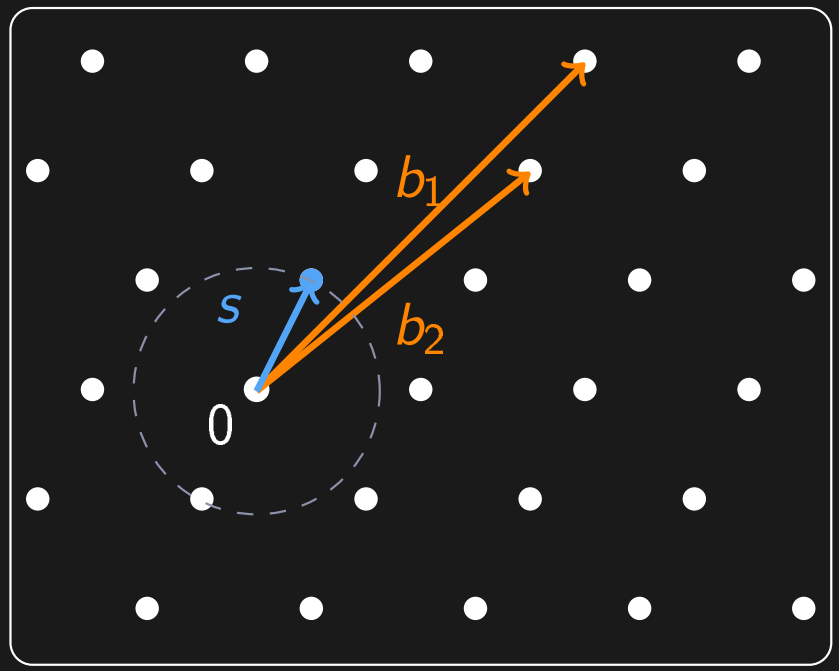
Orange: uniform in $\mathbb{Z}/q\mathbb{Z}$

Blue: uniform in $\{-1, 0, 1\}$

**LWE**: given $A$ and $b$, find $s$

- mostly used in constructions

## Geometric formalism




**SVP**: given $(b_1, b_2)$, find $s$

- mostly used in cryptanalysis

# Two lattice worlds

## Matrix formalism

$$b = A \cdot s + e$$

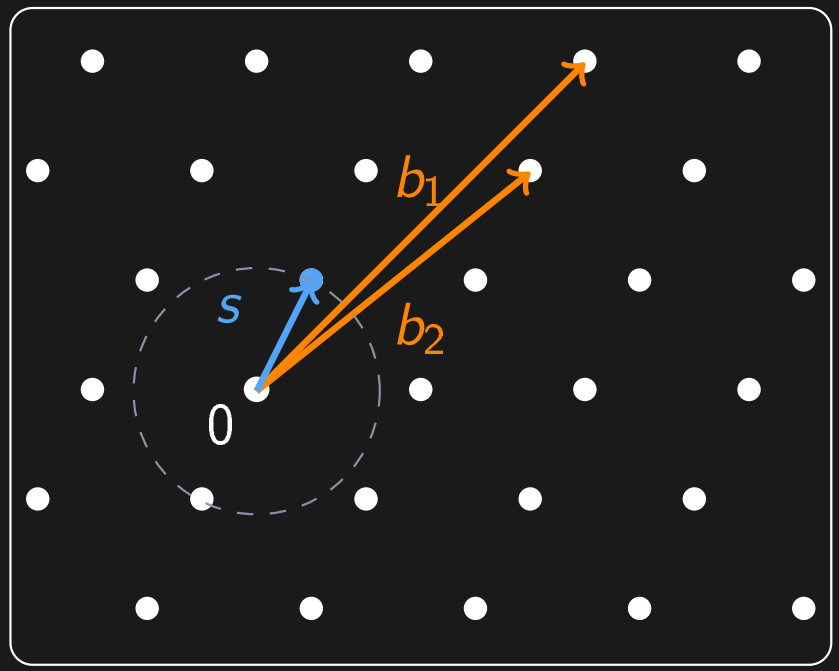
Orange: uniform in $\mathbb{Z}/q\mathbb{Z}$

Blue: uniform in $\{-1, 0, 1\}$

**LWE**: given $A$ and $b$, find $s$

- mostly used in constructions

## Geometric formalism




**SVP**: given $(b_1, b_2)$, find $s$

- mostly used in cryptanalysis

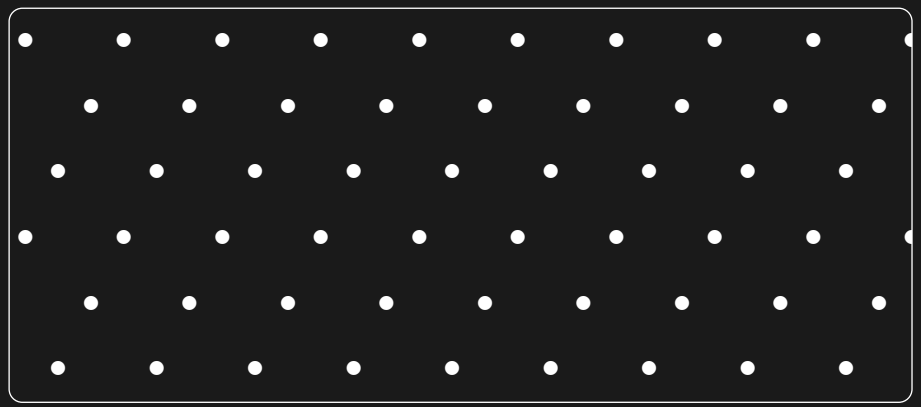
# Plan of the talk

Introduction: Lattices and some definitions

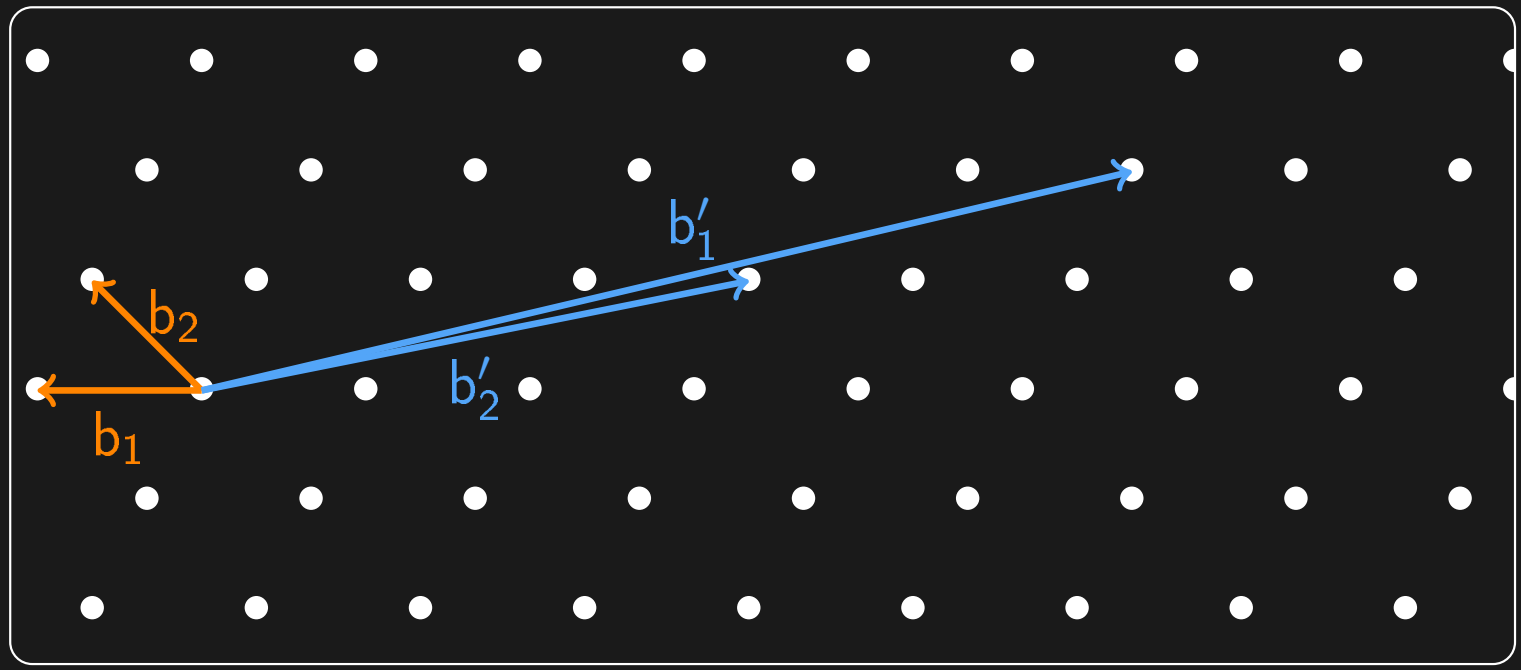
1. Hash-and-sign signatures
2. Attack on a variant of SIS
3. The NTRU problem

} all independent

# Lattices and some definitions



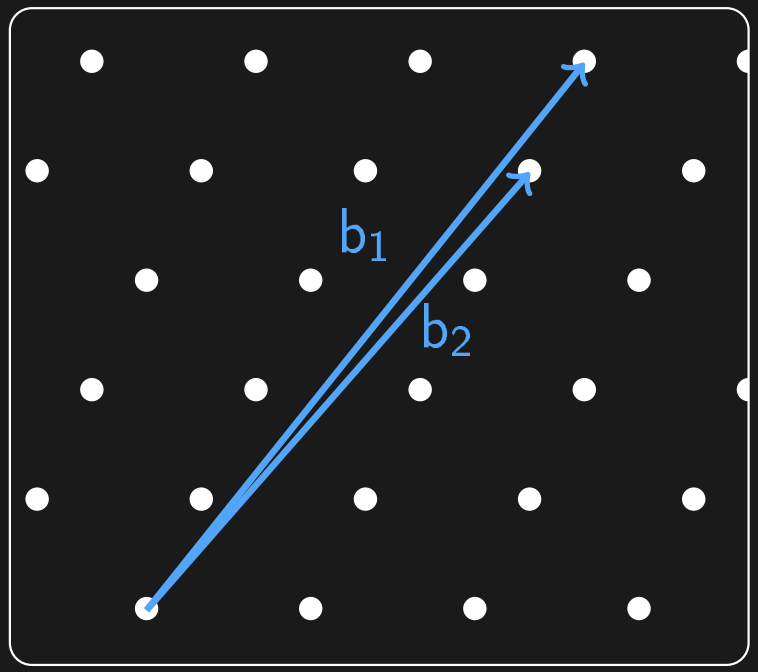
# Lattices



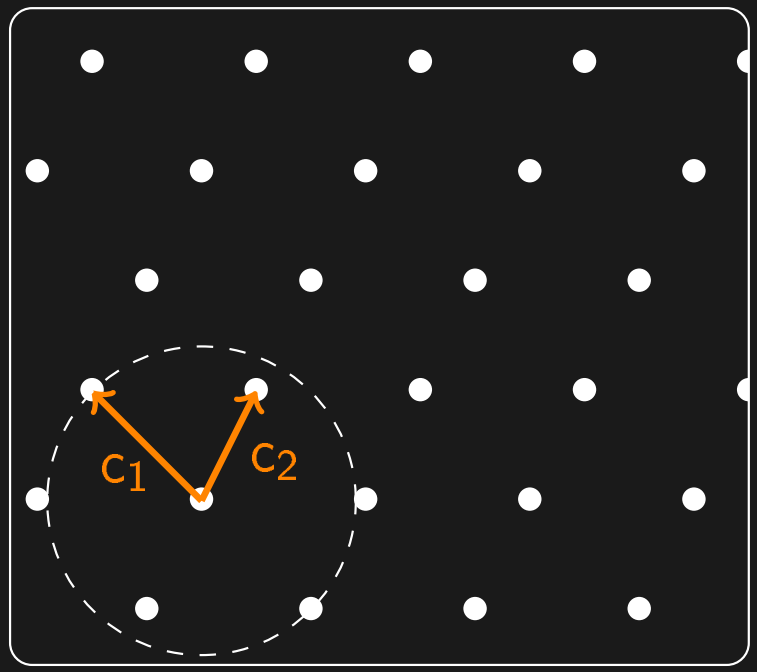

- $\mathcal{L} = \{\sum_{i=1}^{n} x_i b_i \mid \forall i,\, x_i \in \mathbb{Z}\}$ is a lattice
- $(b_1, \ldots, b_n) =: B \in \mathrm{GL}_n(\mathbb{R})$ is a basis (not unique)

# Short basis problem

Input:




Output:

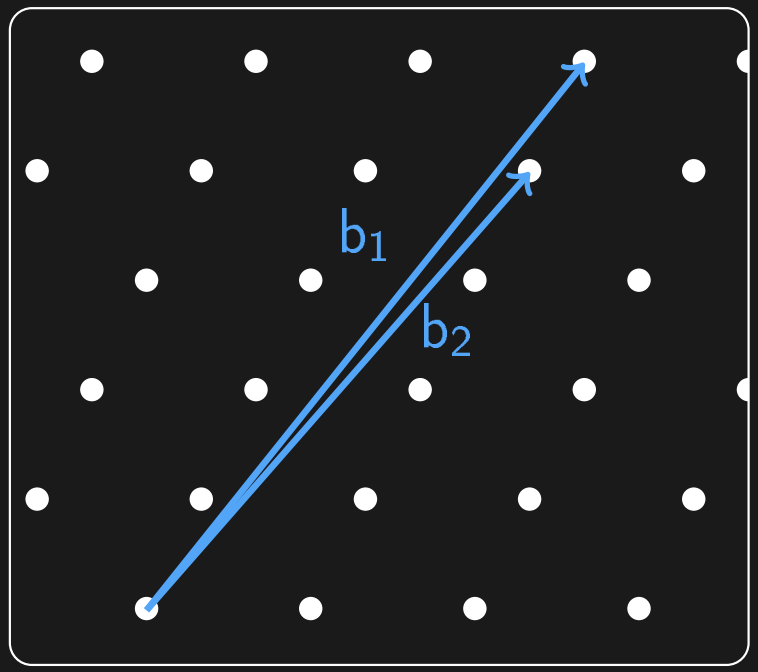



Shortest basis problem

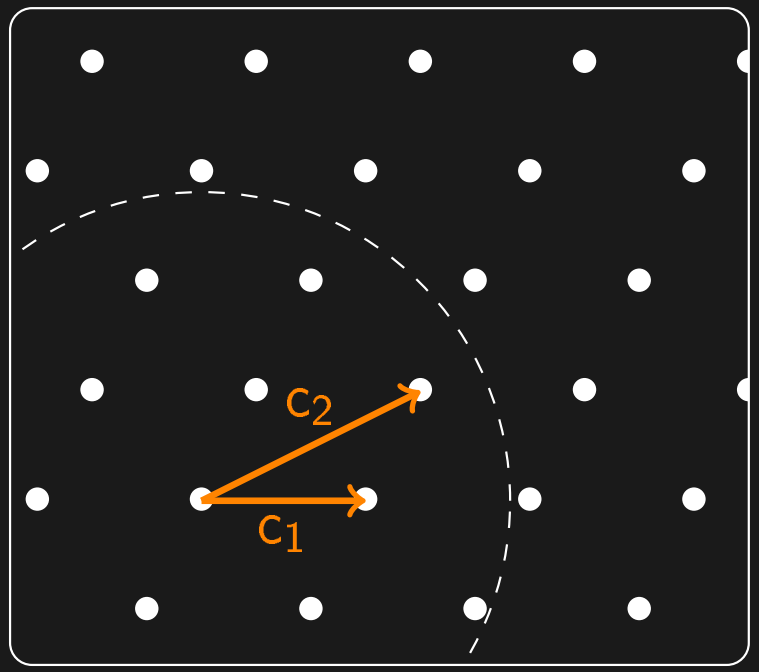
$$\max_i \|c_i\| \leq \min_{B' \text{ basis of } L} \left( \max_i \|b'_i\| \right)$$

# Short basis problem

Input:




Output:

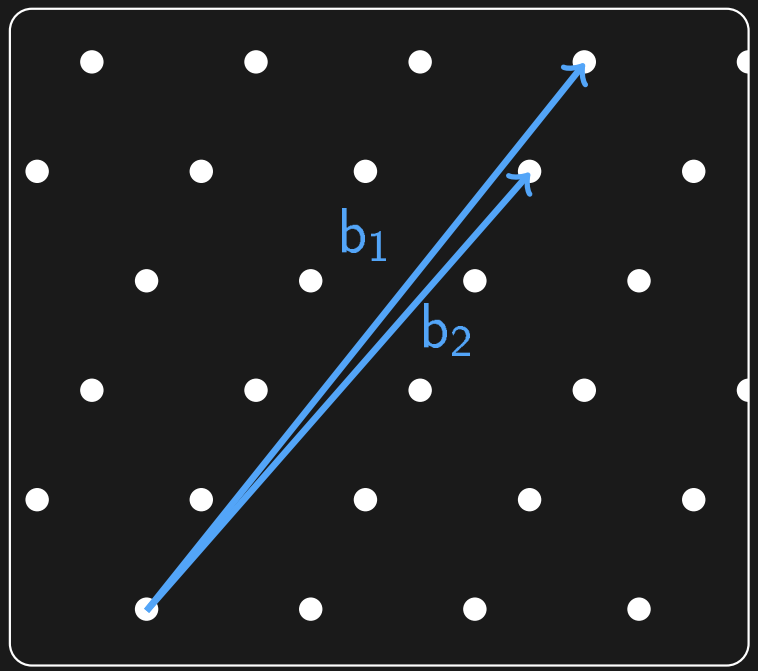



**Approximate** short basis problem

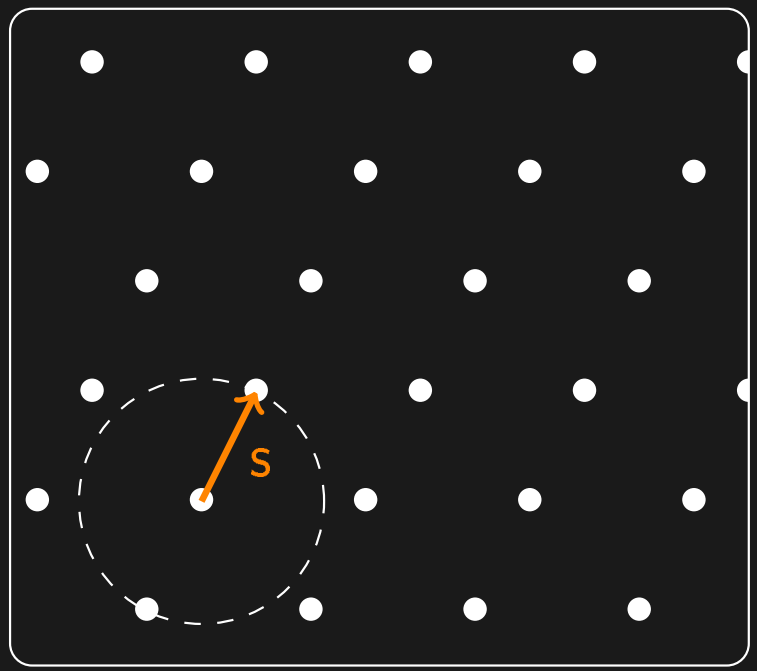
$$\max_i \|c_i\| \leq \gamma \cdot \min_{B' \text{ basis of } L} \left( \max_i \|b'_i\| \right)$$

# Short vector problem

Input:




Output:

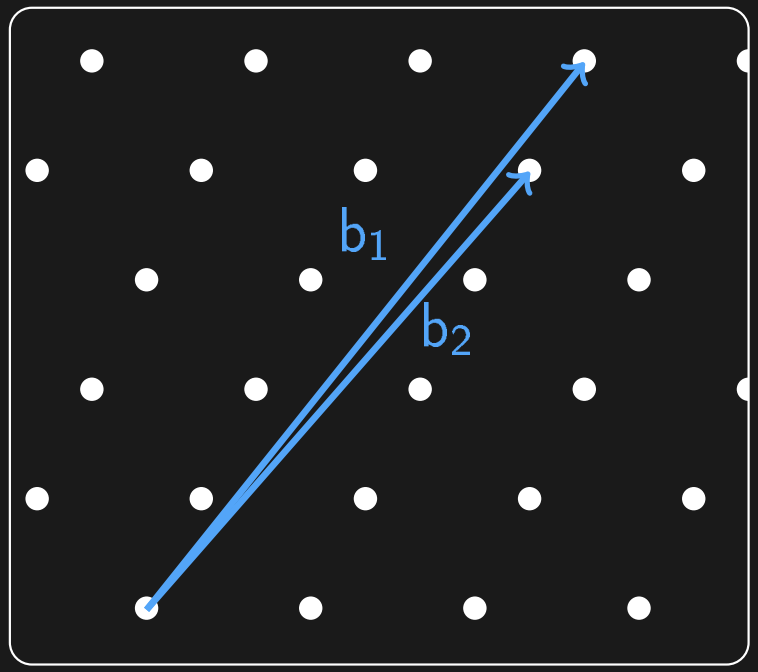



Shortest vector problem

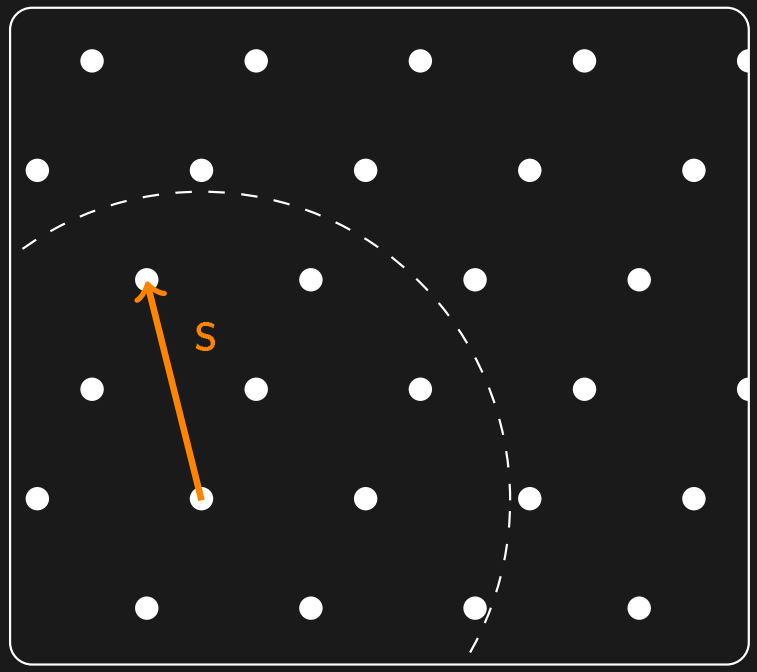
$$\|s\| \leq \min_{v \in \mathcal{L} \setminus \{0\}} \|v\|$$

# Short vector problem

Input:



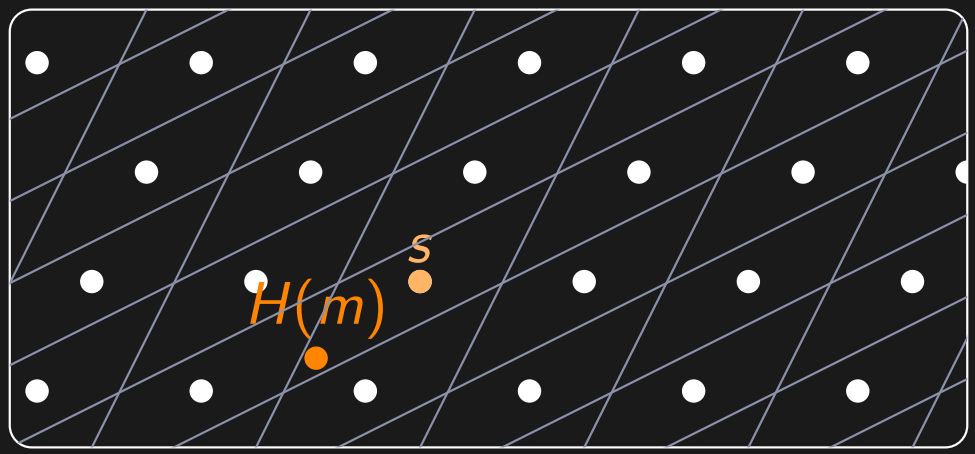

Output:



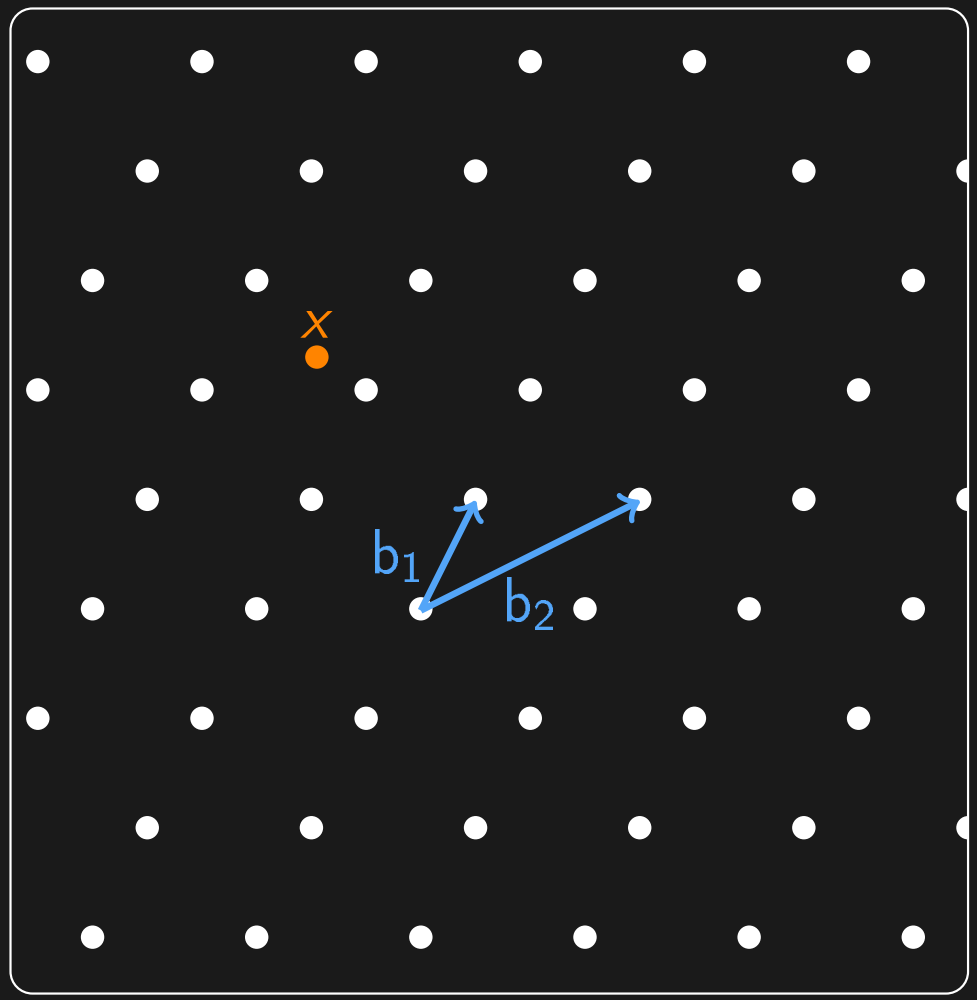

**Approximate** short vector problem

$$\|s\| \leq \gamma \cdot \min_{v \in \mathcal{L} \setminus \{0\}} \|v\|$$

# Hash-and-sign signatures



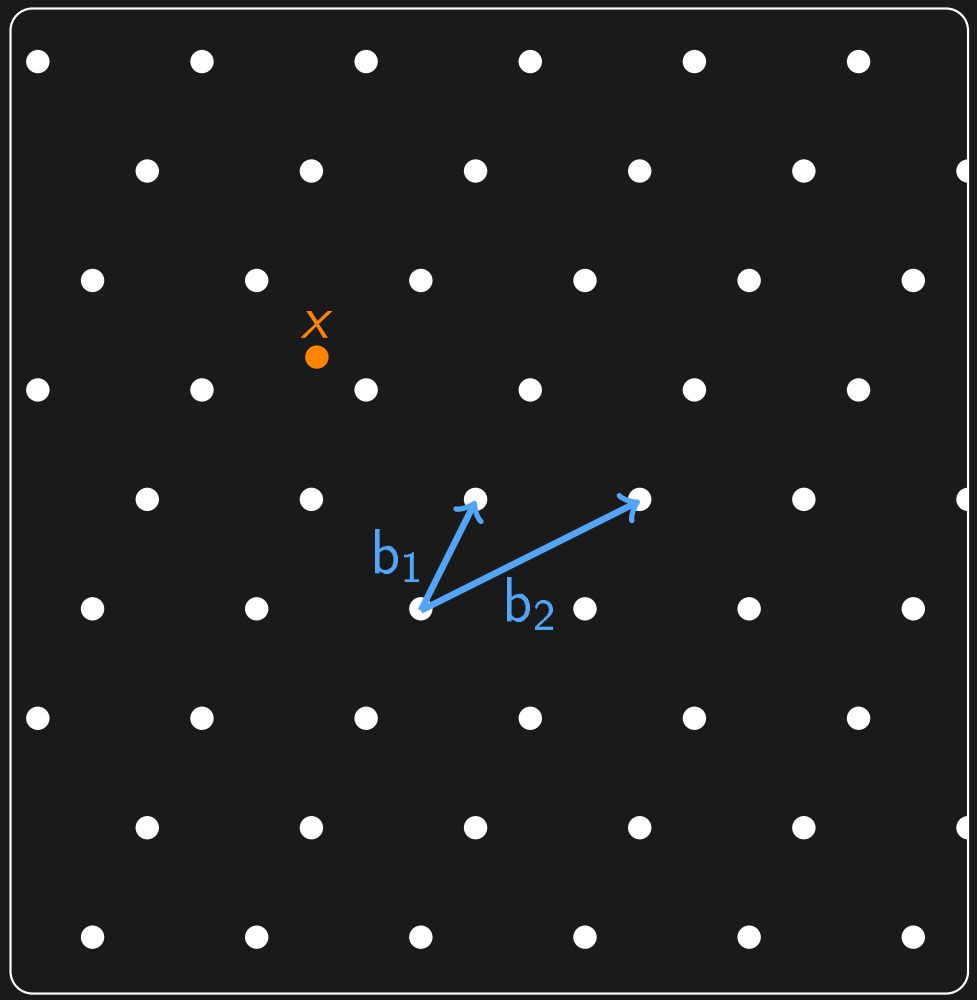
# Finding a close vector using a short basis




Input: $x = 3.7 \cdot \mathbf{b}_1 - 1.4 \cdot \mathbf{b}_2$

Objective: find $s \in \mathcal{L}$ close to $x$

# Finding a close vector using a short basis

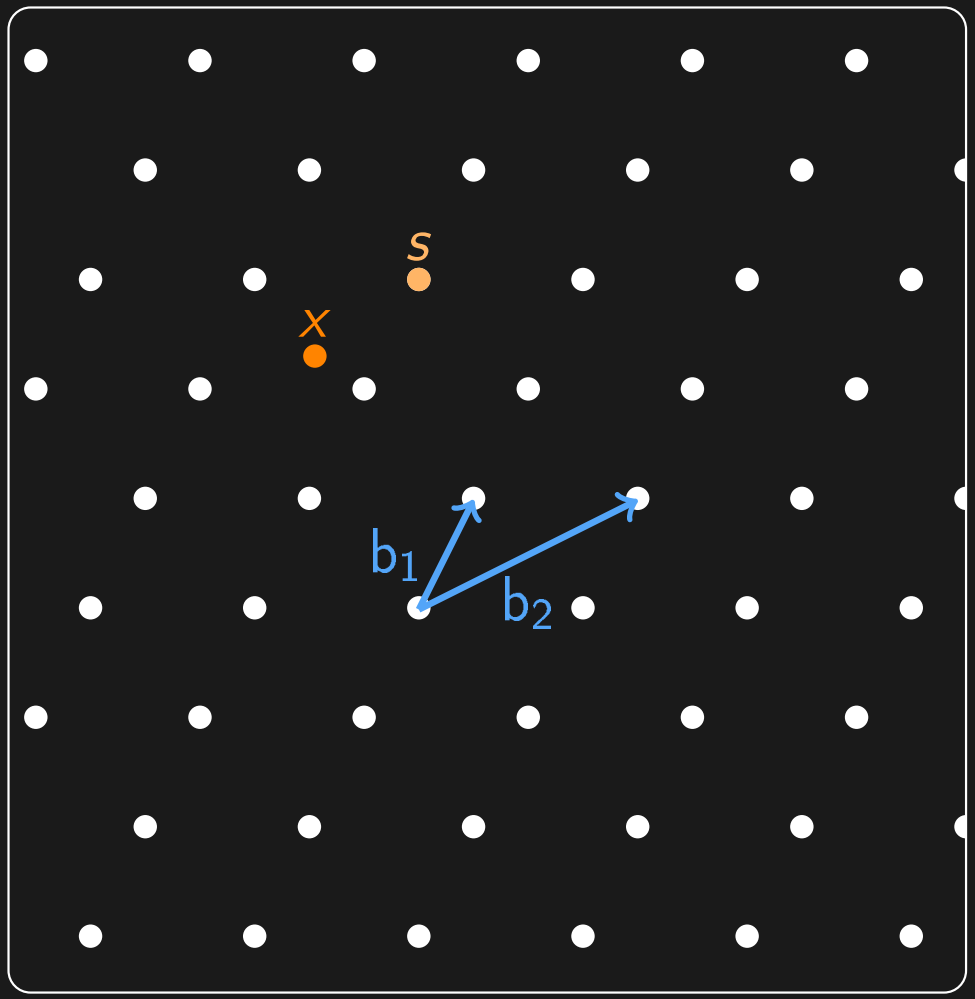



Input: $x = 3.7 \cdot b_1 - 1.4 \cdot b_2$

Objective: find $s \in \mathcal{L}$ close to $x$

Algo: round each coordinate

# Finding a close vector using a short basis



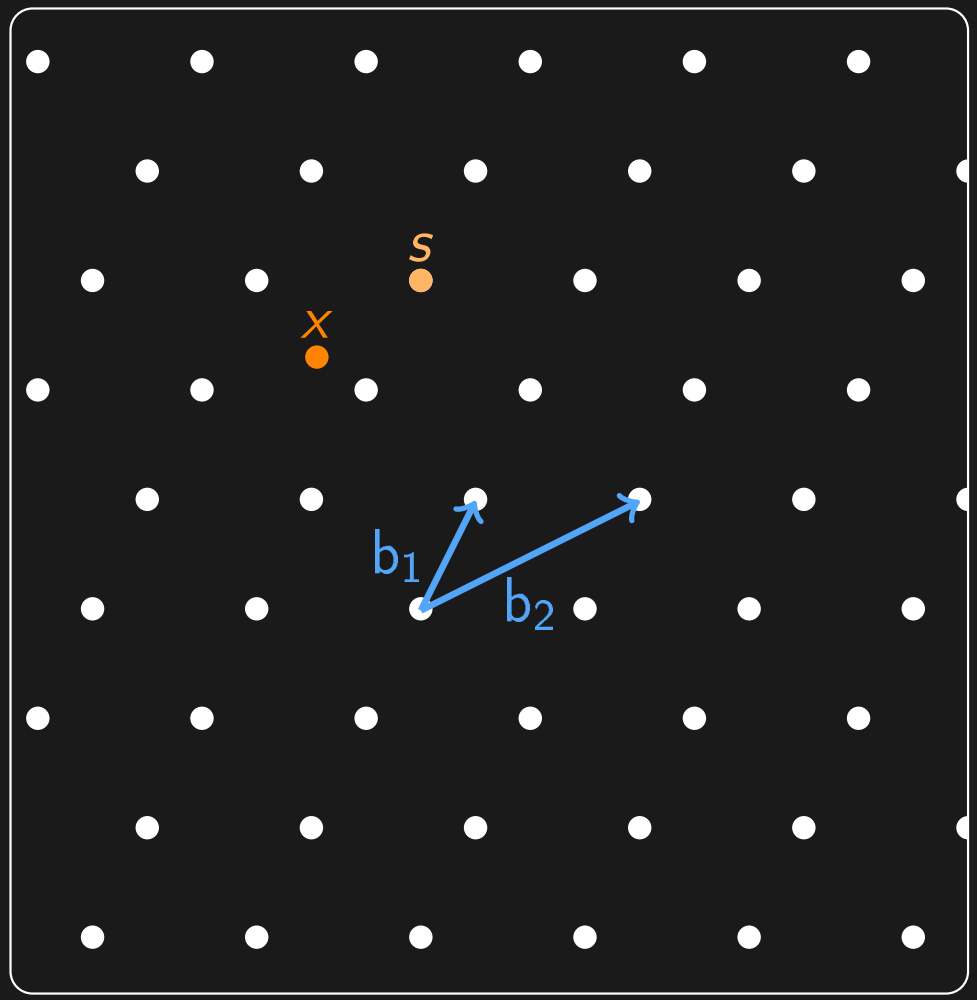

Input: $x = 3.7 \cdot b_1 - 1.4 \cdot b_2$

Objective: find $s \in \mathcal{L}$ close to $x$

Algo: round each coordinate

Output: $s = 4 \cdot b_1 - 1 \cdot b_2$

# Finding a close vector using a short basis




Input: $x = 3.7 \cdot b_1 - 1.4 \cdot b_2$

Objective: find $s \in \mathcal{L}$ close to $x$

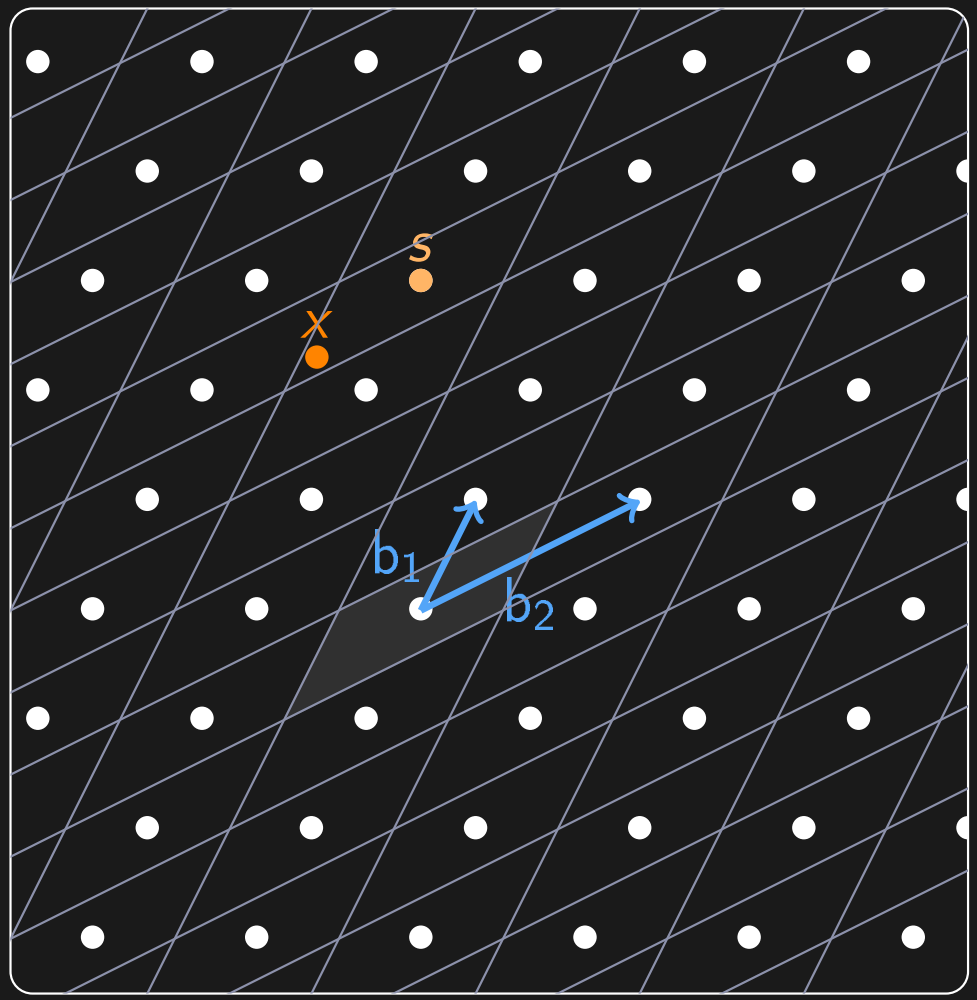
Algo: round each coordinate

Output: $s = 4 \cdot b_1 - 1 \cdot b_2$

The smaller the basis, the closer the solution

(called Babai's round-off algorithm)

# Finding a close vector using a short basis




Input: $x = 3.7 \cdot b_1 - 1.4 \cdot b_2$

Objective: find $s \in \mathcal{L}$ close to $x$

Algo: round each coordinate

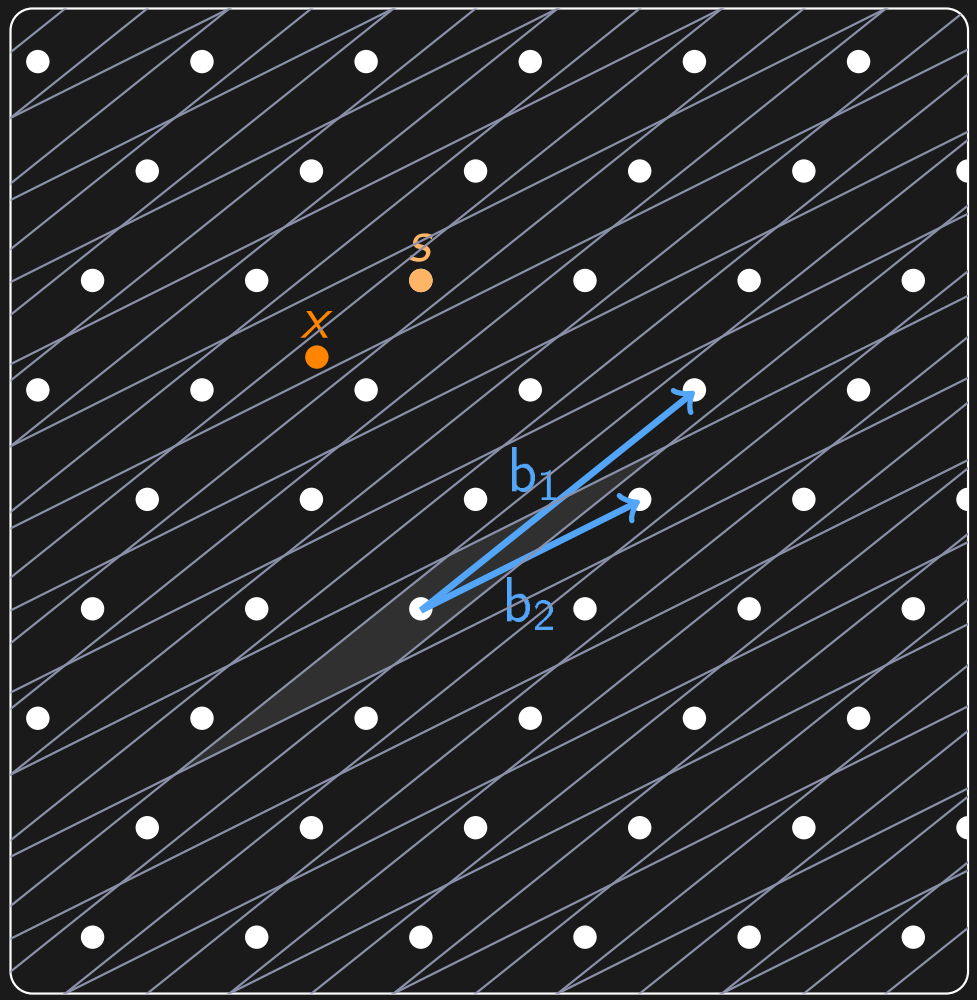
Output: $s = 4 \cdot b_1 - 1 \cdot b_2$

The smaller the basis, the closer the solution

(called Babai's round-off algorithm)

$$= \left\{ x_1 b_1 + x_2 b_2 \,\middle|\, |x_i| \leq \tfrac{1}{2} \right\}$$

# Finding a close vector using a short basis




Input: $x = 3.7 \cdot b_1 - 1.4 \cdot b_2$

Objective: find $s \in \mathcal{L}$ close to $x$

Algo: round each coordinate

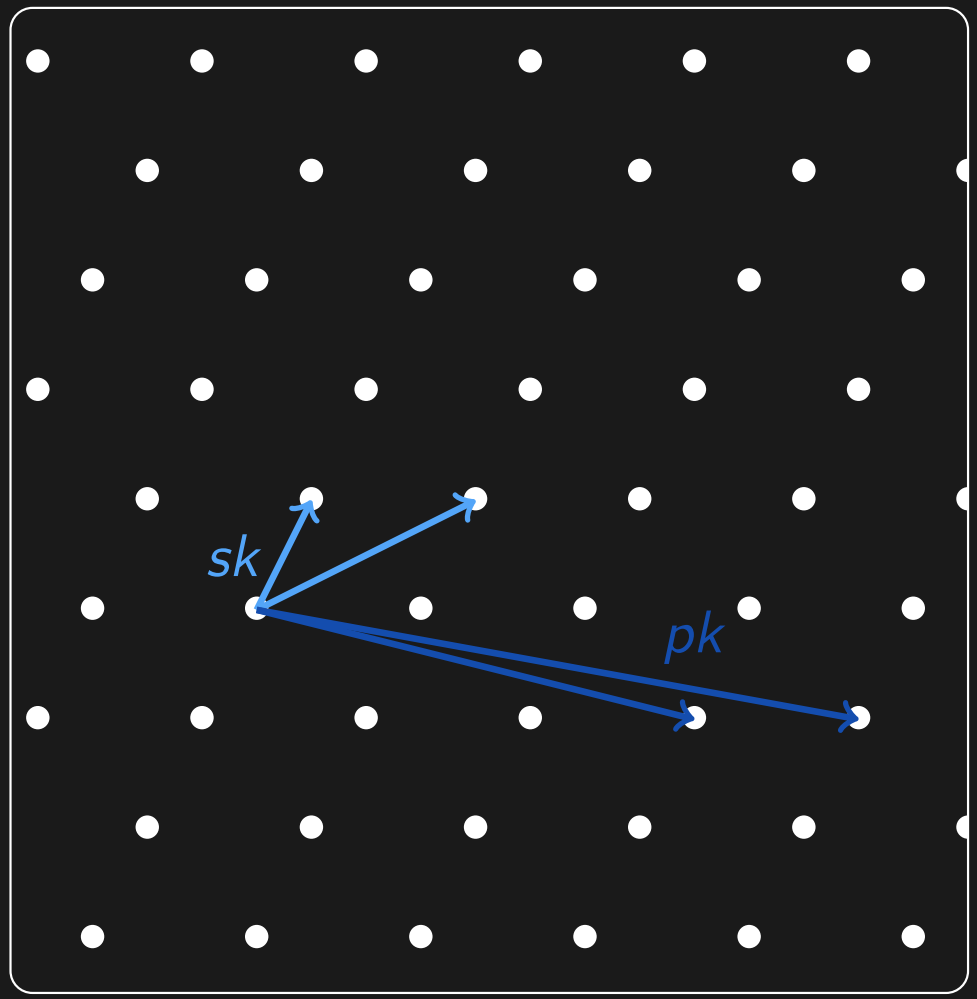
Output: $s = 4 \cdot b_1 - 1 \cdot b_2$

The smaller the basis, the closer the solution

(called Babai's round-off algorithm)

 $= \left\{ x_1 b_1 + x_2 b_2 \,\middle|\, |x_i| \leq \frac{1}{2} \right\}$

# Hash-and-sign: first idea [GGH97]

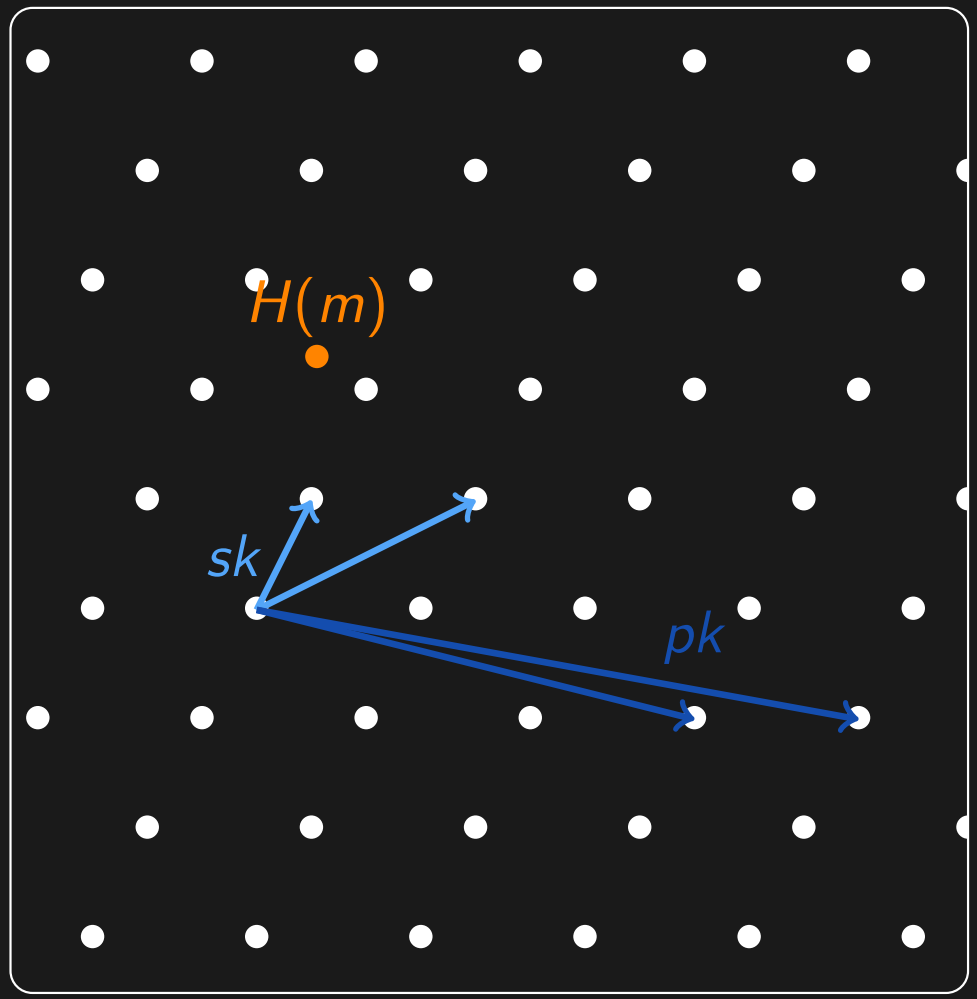



KeyGen:

- $pk$ = bad basis of $\mathcal{L}$
- $sk$ = short basis of $\mathcal{L}$

[GGH97] Goldreich, Goldwasser, and Halevi. Public-key cryptosystems from lattice reduction problems. CRYPTO

# Hash-and-sign: first idea [GGH97]




KeyGen:

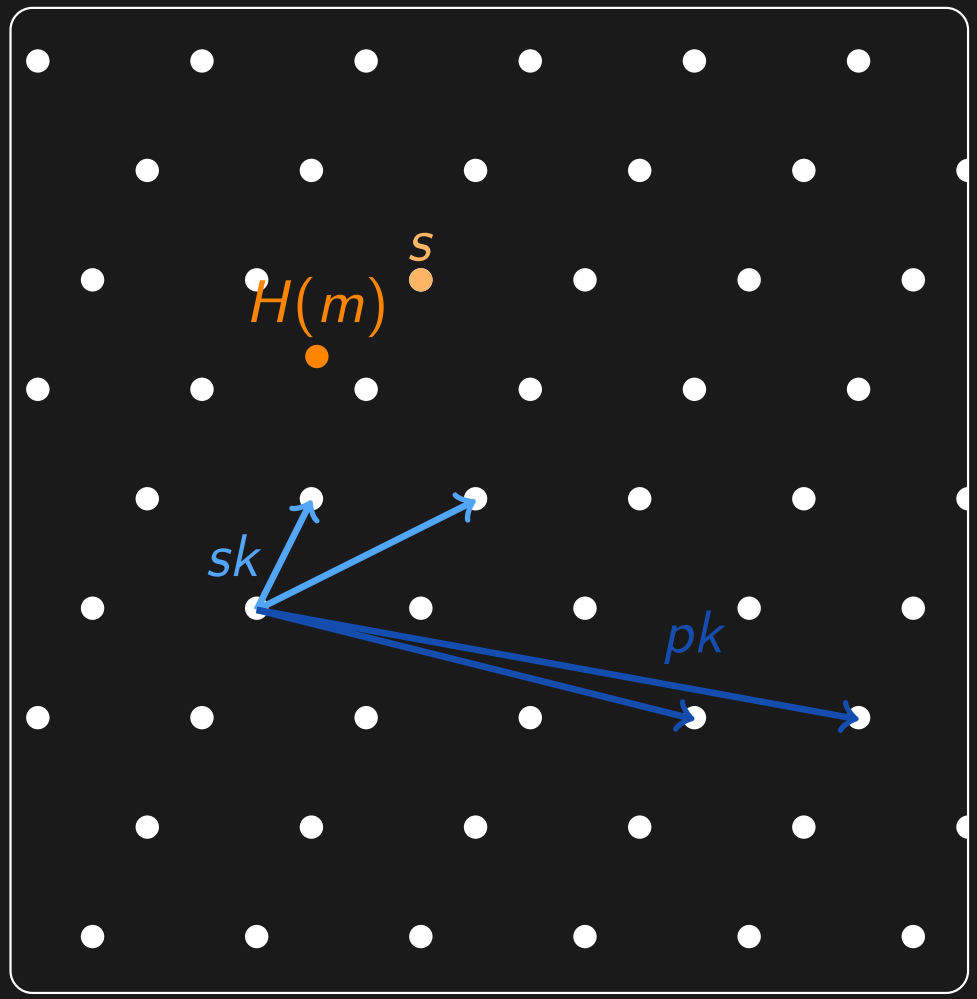
- $pk$ = bad basis of $\mathcal{L}$
- $sk$ = short basis of $\mathcal{L}$

Sign$(m, sk)$:

- $x = H(m)$ (hash the message)

[GGH97] Goldreich, Goldwasser, and Halevi. Public-key cryptosystems from lattice reduction problems. CRYPTO

# Hash-and-sign: first idea [GGH97]




KeyGen:

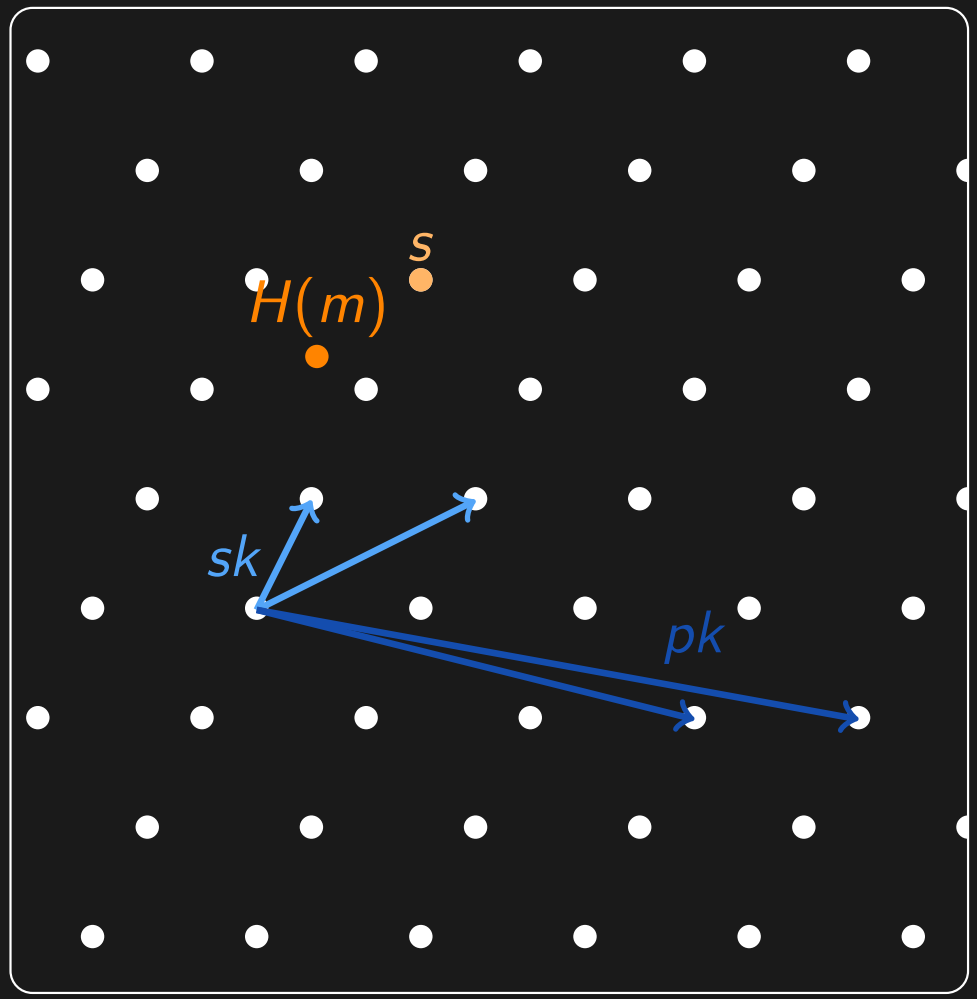
- $pk$ = bad basis of $\mathcal{L}$
- $sk$ = short basis of $\mathcal{L}$

Sign$(m, sk)$:

- $x = H(m)$ (hash the message)
- output $s \in \mathcal{L}$ close to $H(m)$

[GGH97] Goldreich, Goldwasser, and Halevi. Public-key cryptosystems from lattice reduction problems. CRYPTO

# Hash-and-sign: first idea [GGH97]




KeyGen:

- $pk$ = bad basis of $\mathcal{L}$
- $sk$ = short basis of $\mathcal{L}$

Sign$(m, sk)$:

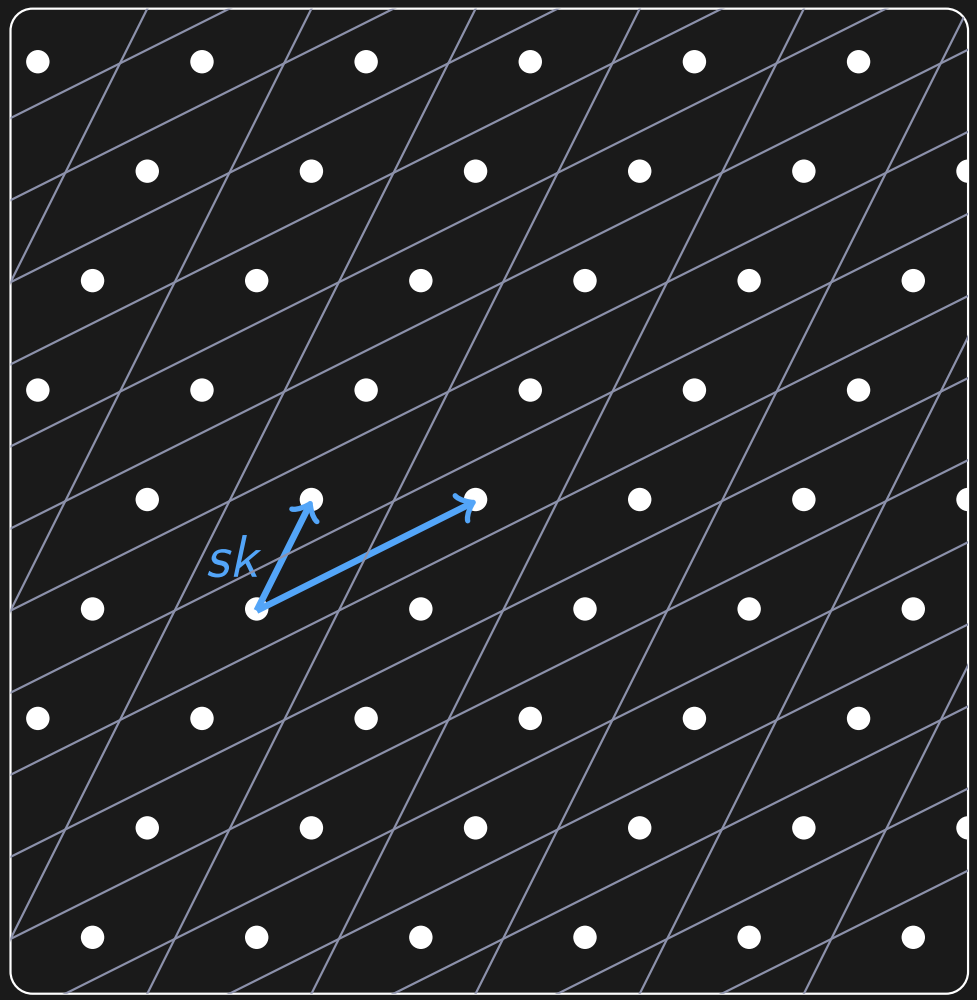
- $x = H(m)$ (hash the message)
- output $s \in \mathcal{L}$ close to $H(m)$

Verify$(m, s, pk)$:

- check that $s \in \mathcal{L}$
- check that $H(m) - s$ is small

[GGH97] Goldreich, Goldwasser, and Halevi. Public-key cryptosystems from lattice reduction problems. CRYPTO

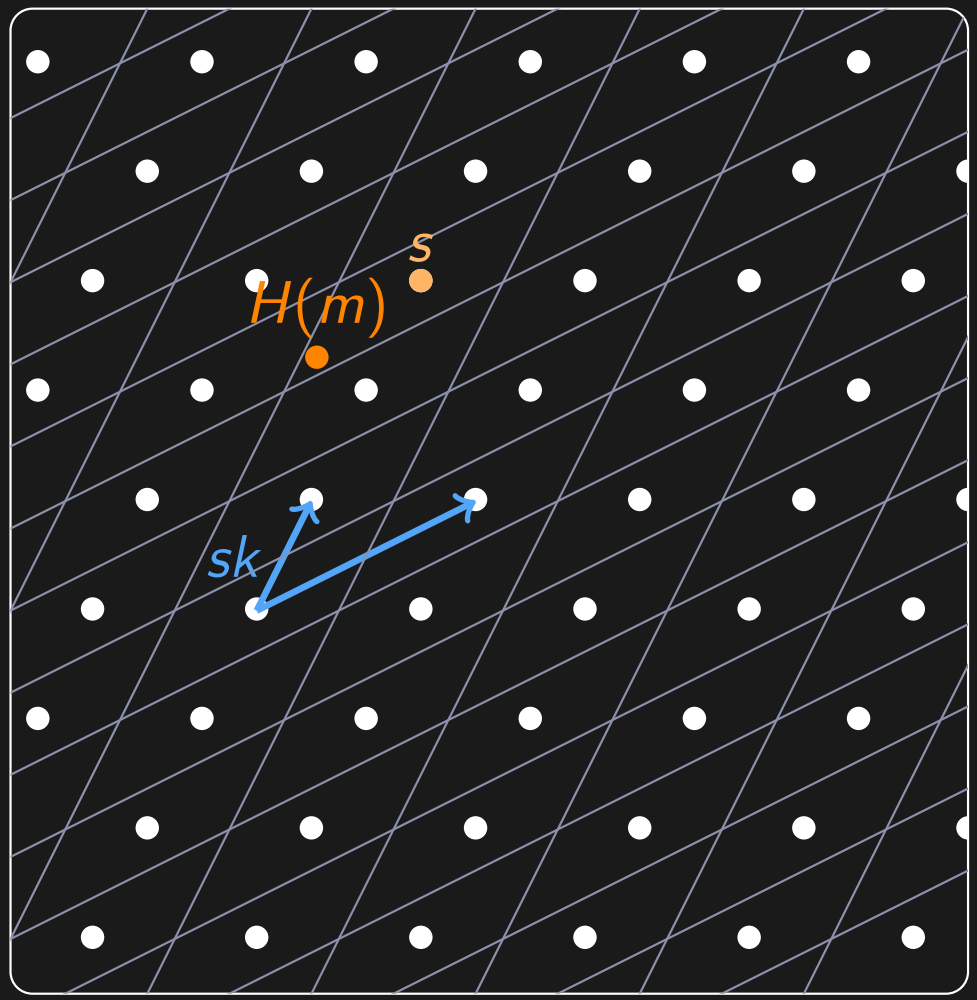
# Attack on this first idea [NR06]




Parallelepiped attack:

[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology

# Attack on this first idea [NR06]

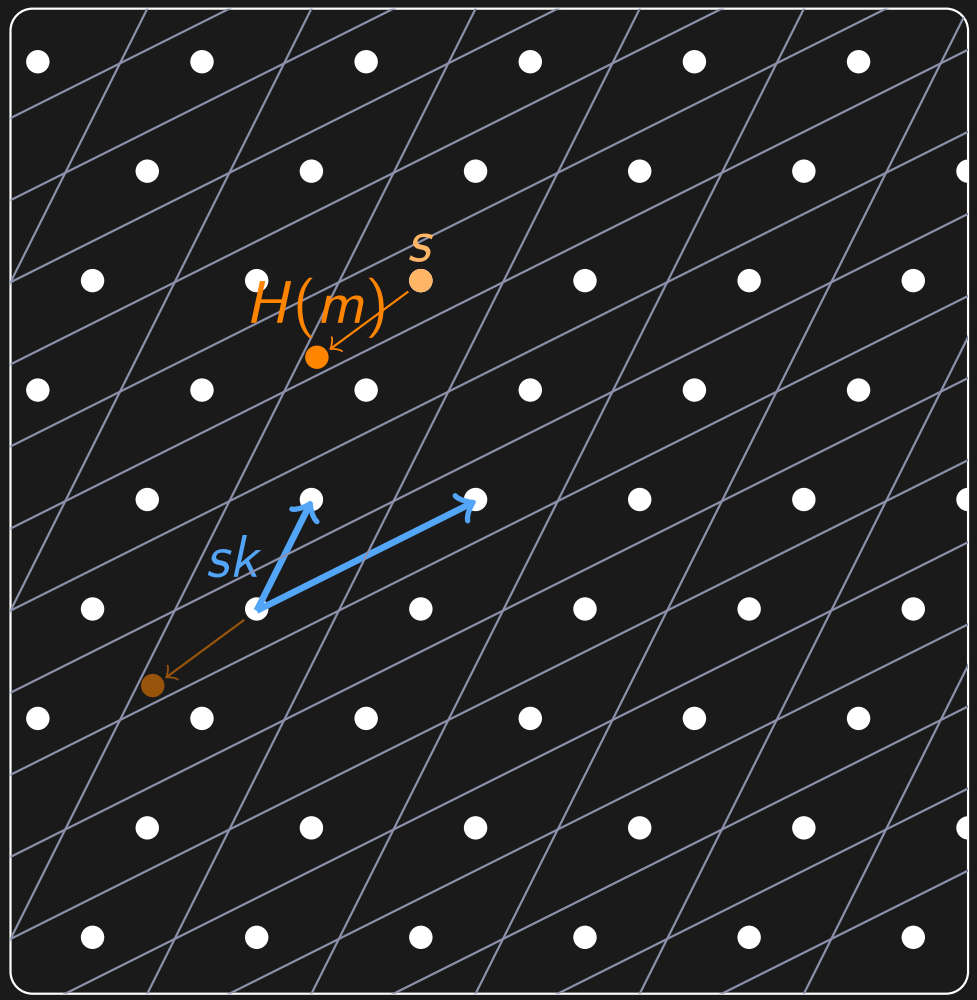



Parallelepiped attack:

- ask for a signature $s$ on $m$

[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology

# Attack on this first idea [NR06]

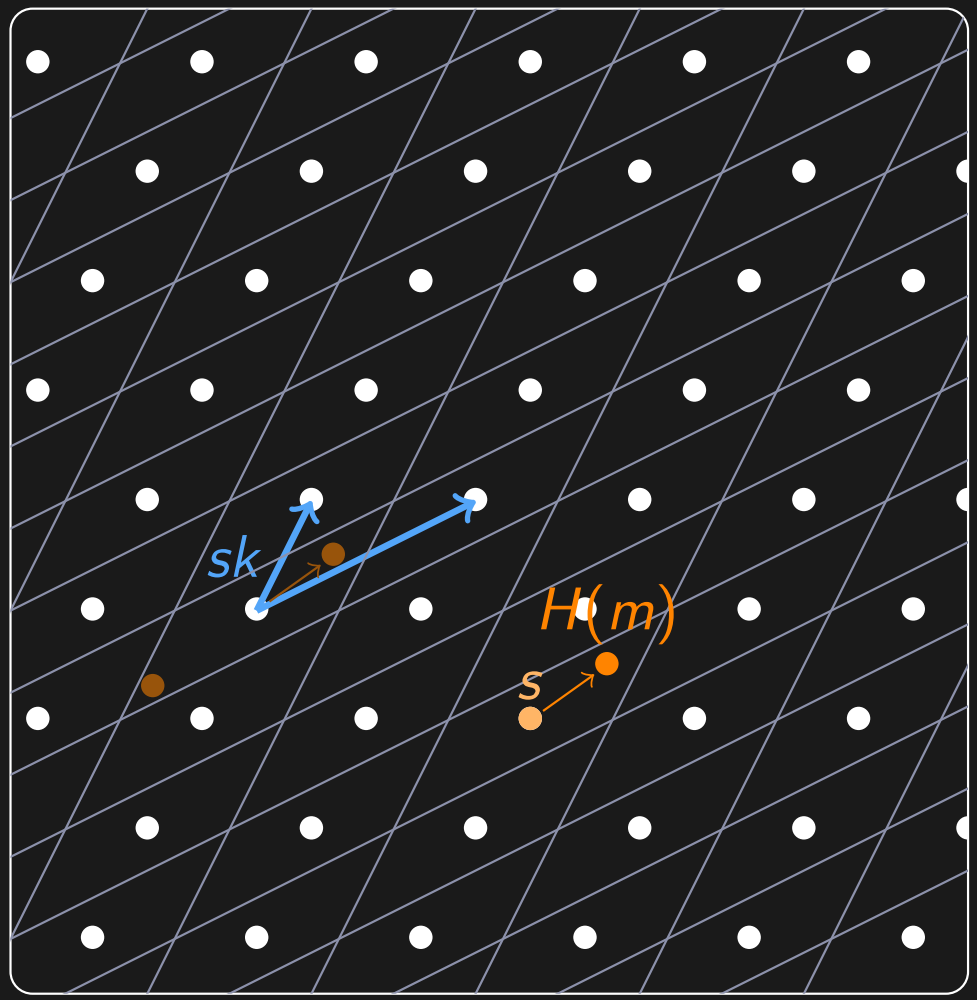



Parallelepiped attack:

- ask for a signature $s$ on $m$
- plot $H(m) - s$

[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology

# Attack on this first idea [NR06]

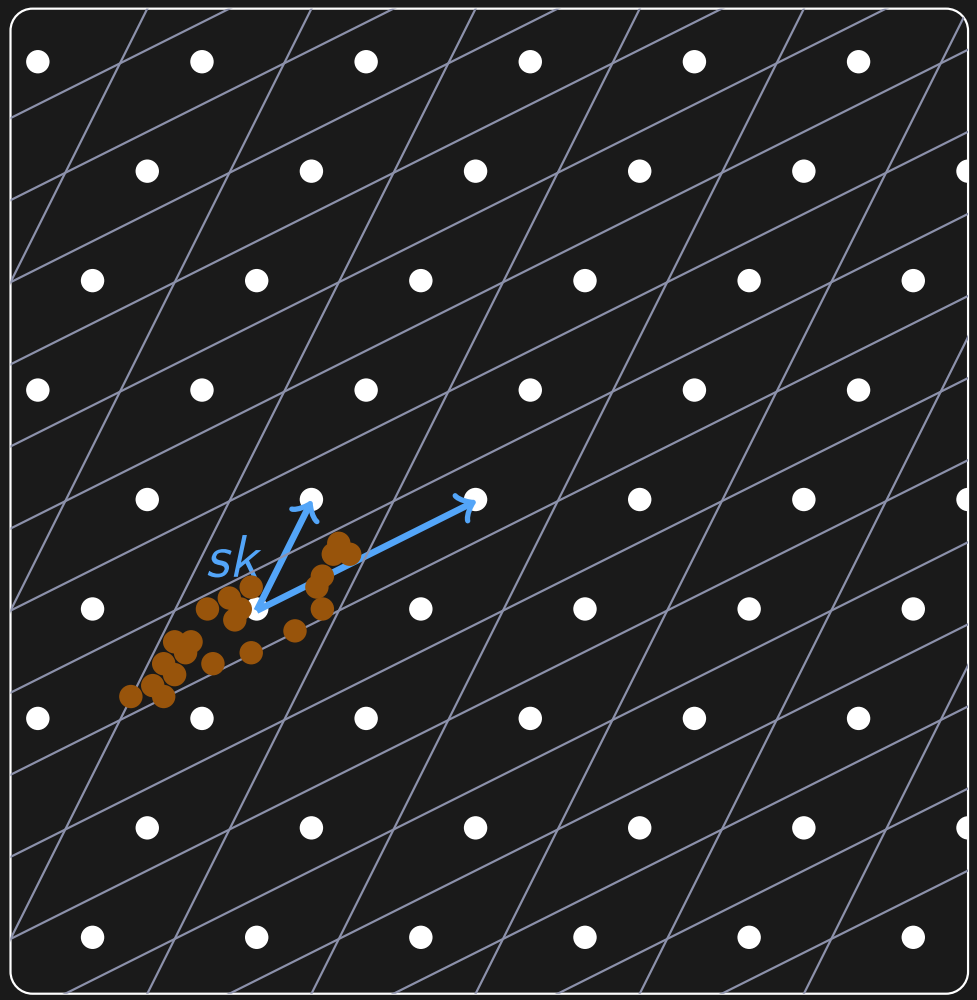



Parallelepiped attack:

- ask for a signature $s$ on $m$
- plot $H(m) - s$
- repeat

[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology

# Attack on this first idea [NR06]

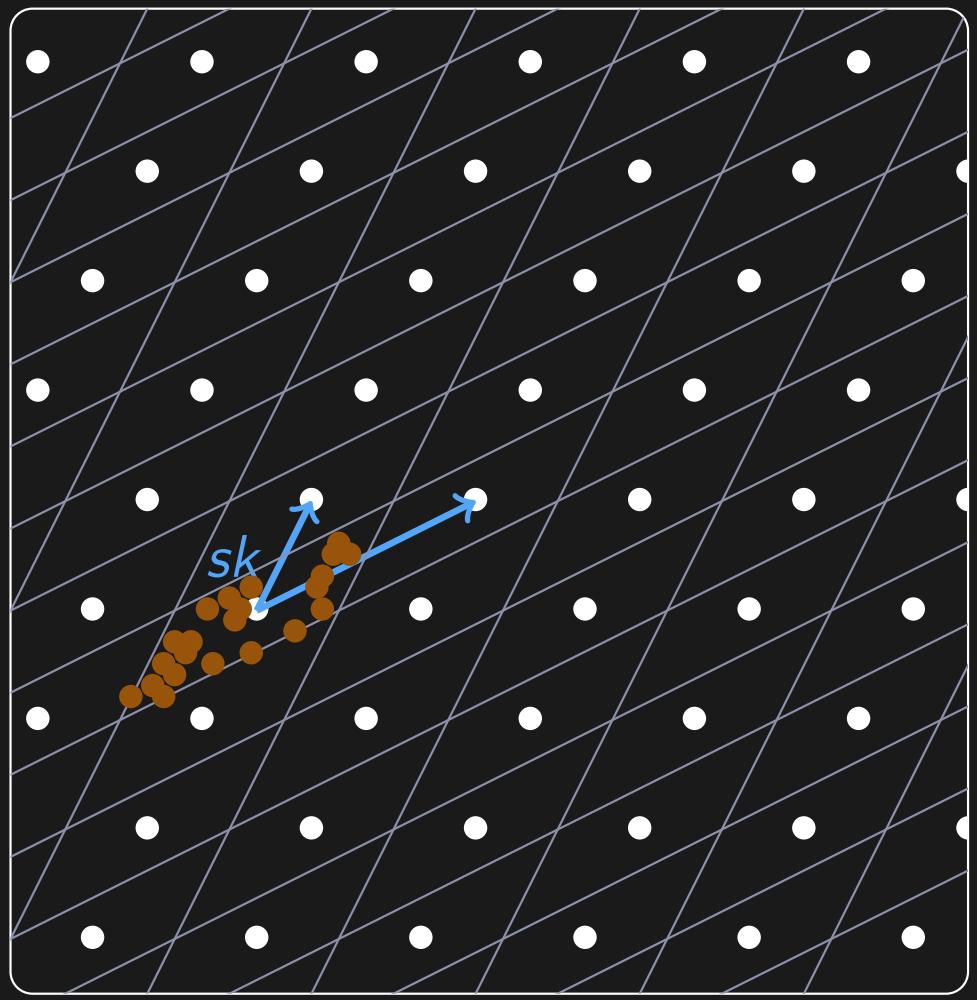



Parallelepiped attack:

- ask for a signature $s$ on $m$
- plot $H(m) - s$
- repeat

[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology

# Attack on this first idea [NR06]



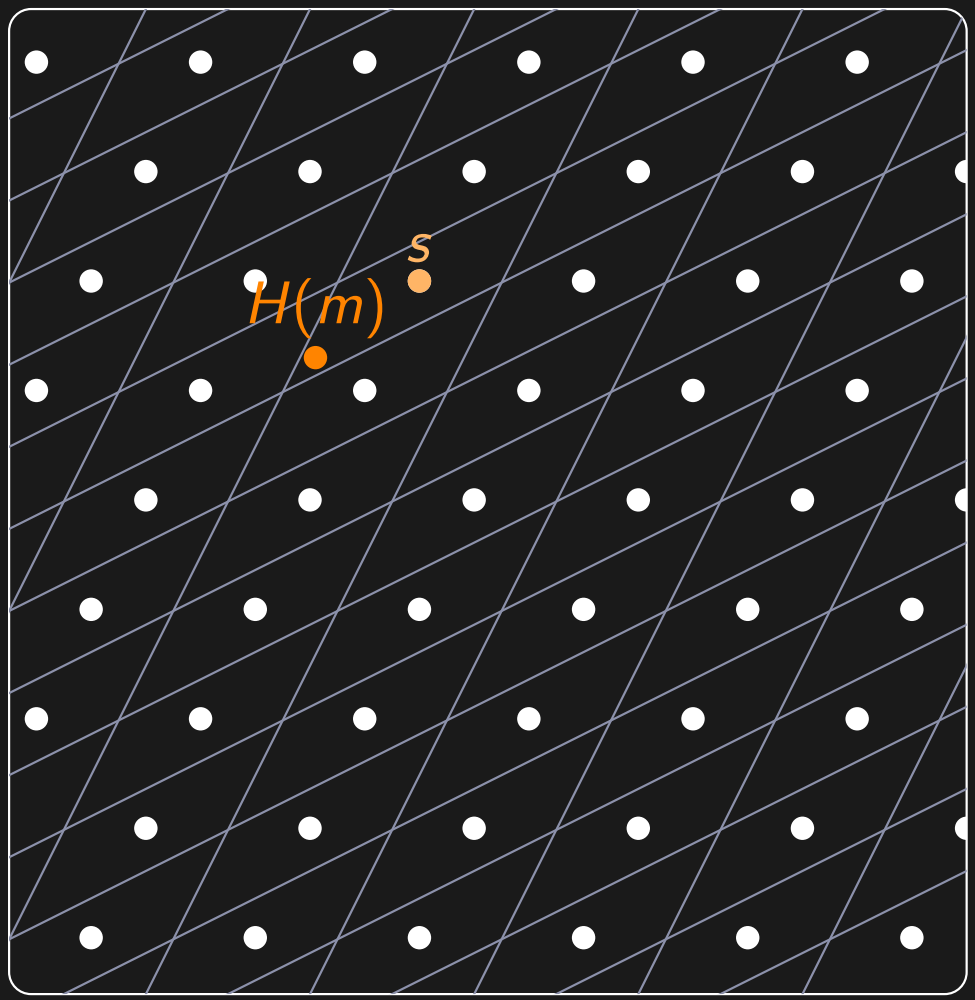

Parallelepiped attack:

- ask for a signature $s$ on $m$
- plot $H(m) - s$
- repeat

From the shape of the parallelepiped, one can recover the short basis

[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology

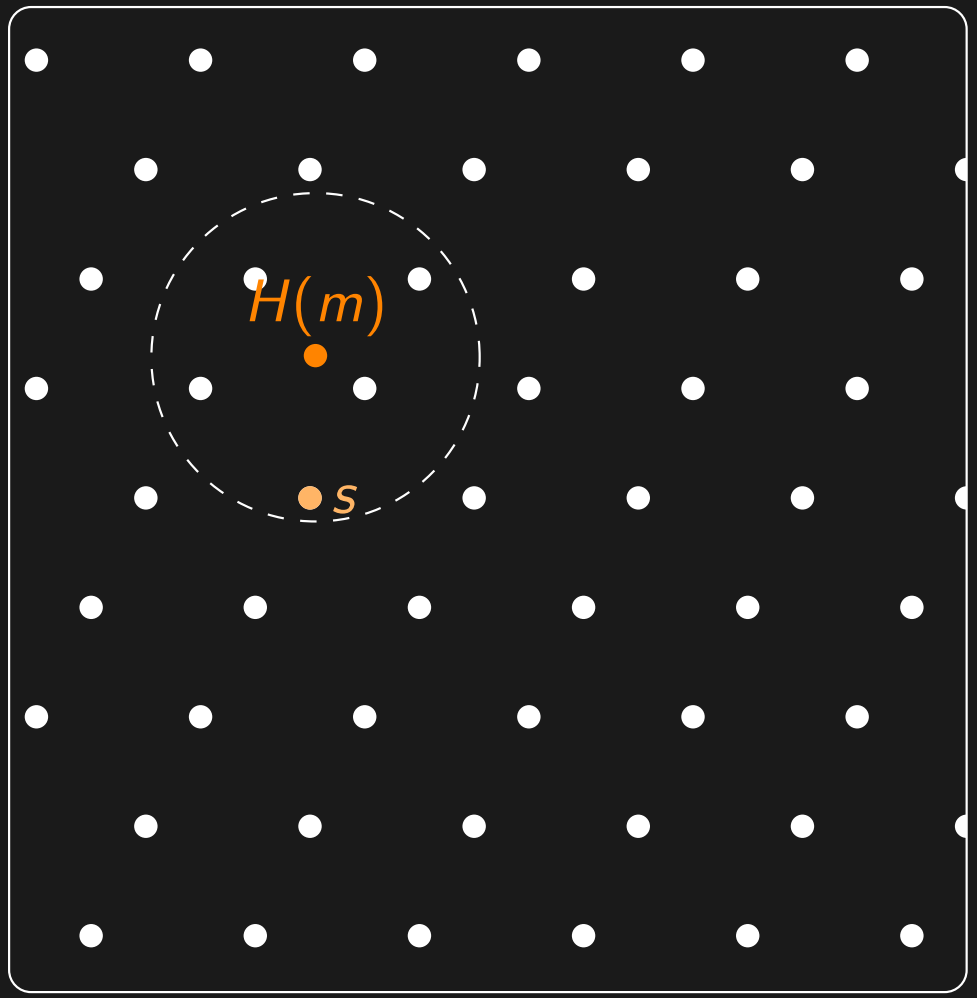
# Preventing the attack [GPV08]




Idea: do not decode deterministically but randomly

[GPV08] Gentry, Peikert, Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC.

# Preventing the attack [GPV08]



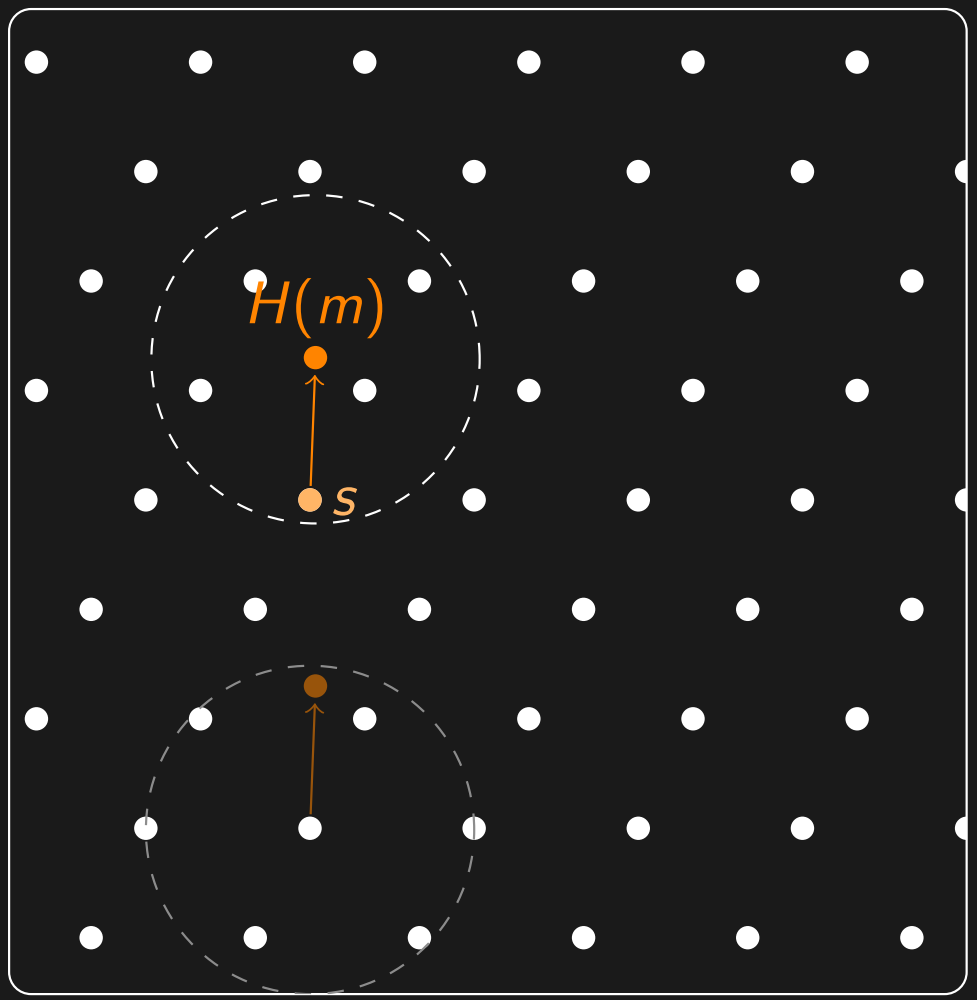

Idea: do not decode deterministically but randomly

Sign$(m, sk)$:

- $x = H(m)$ (hash the message)
- sample $s \in \mathcal{L} \cap \mathcal{B}_r(x)$
  (small radius $r \approx \max_i \|b_i\|$,
  thanks to short basis $b_1, \ldots, b_n$)

[GPV08] Gentry, Peikert, Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC.

# Preventing the attack [GPV08]



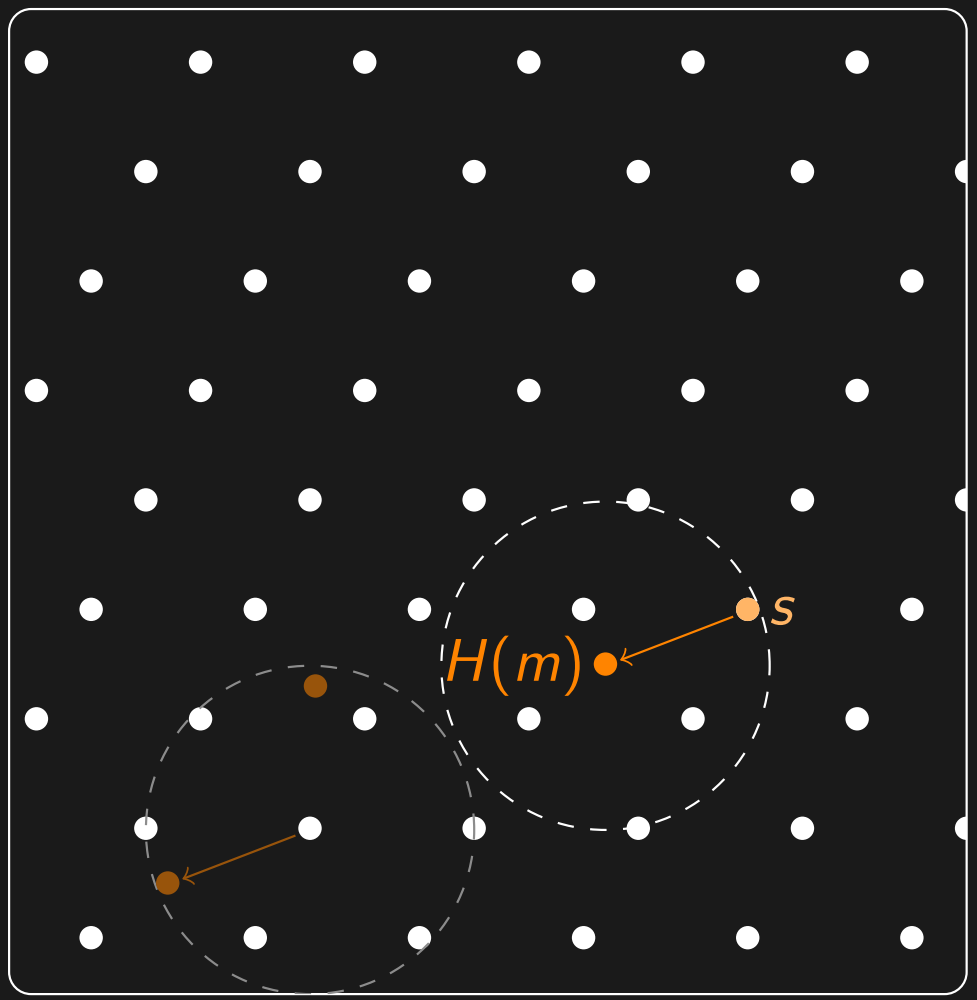

Idea: do not decode deterministically but randomly

Sign$(m, sk)$:

- $x = H(m)$ (hash the message)
- sample $s \in \mathcal{L} \cap \mathcal{B}_r(x)$
  (small radius $r \approx \max_i \|b_i\|$,
  thanks to short basis $b_1, \ldots, b_n$)

[GPV08] Gentry, Peikert, Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC.

# Preventing the attack [GPV08]



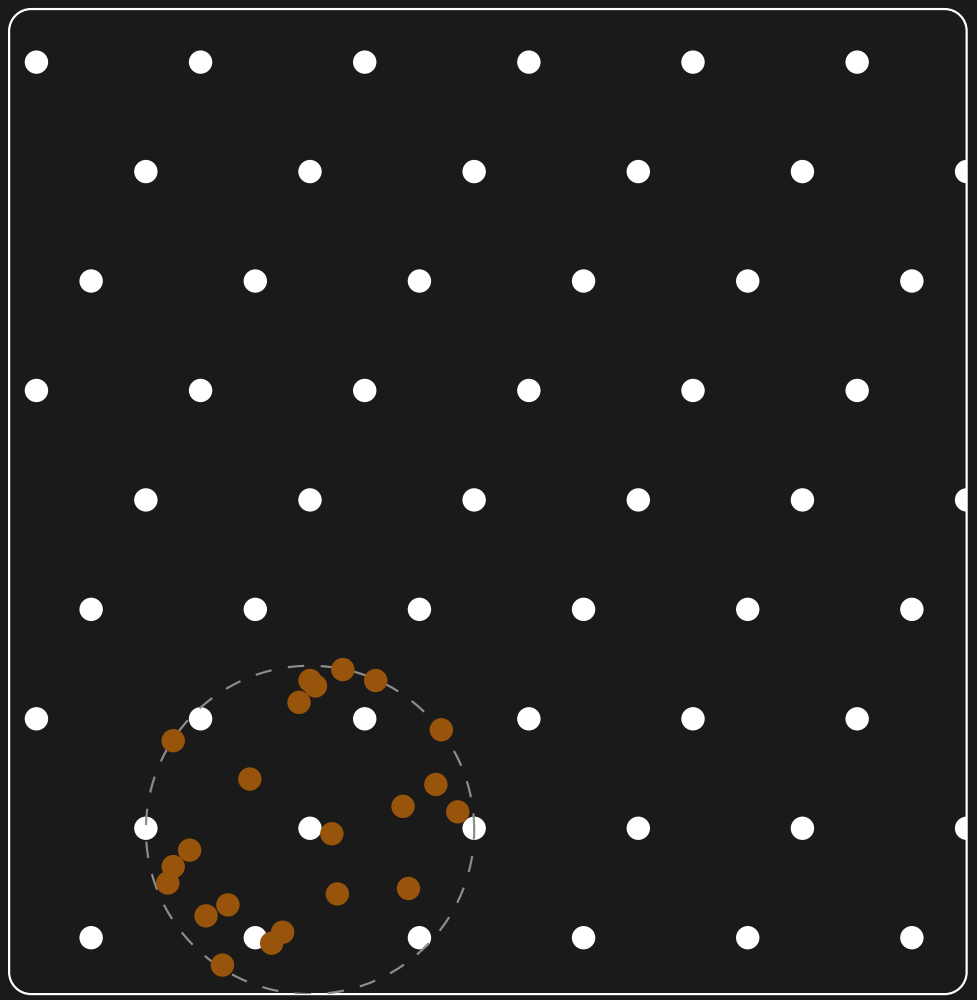

Idea: do not decode deterministically but randomly

Sign$(m, sk)$:

- $x = H(m)$ (hash the message)
- sample $s \in \mathcal{L} \cap \mathcal{B}_r(x)$
  (small radius $r \approx \max_i \|b_i\|$,
  thanks to short basis $b_1, \ldots, b_n$)

[GPV08] Gentry, Peikert, Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC.

# Preventing the attack [GPV08]



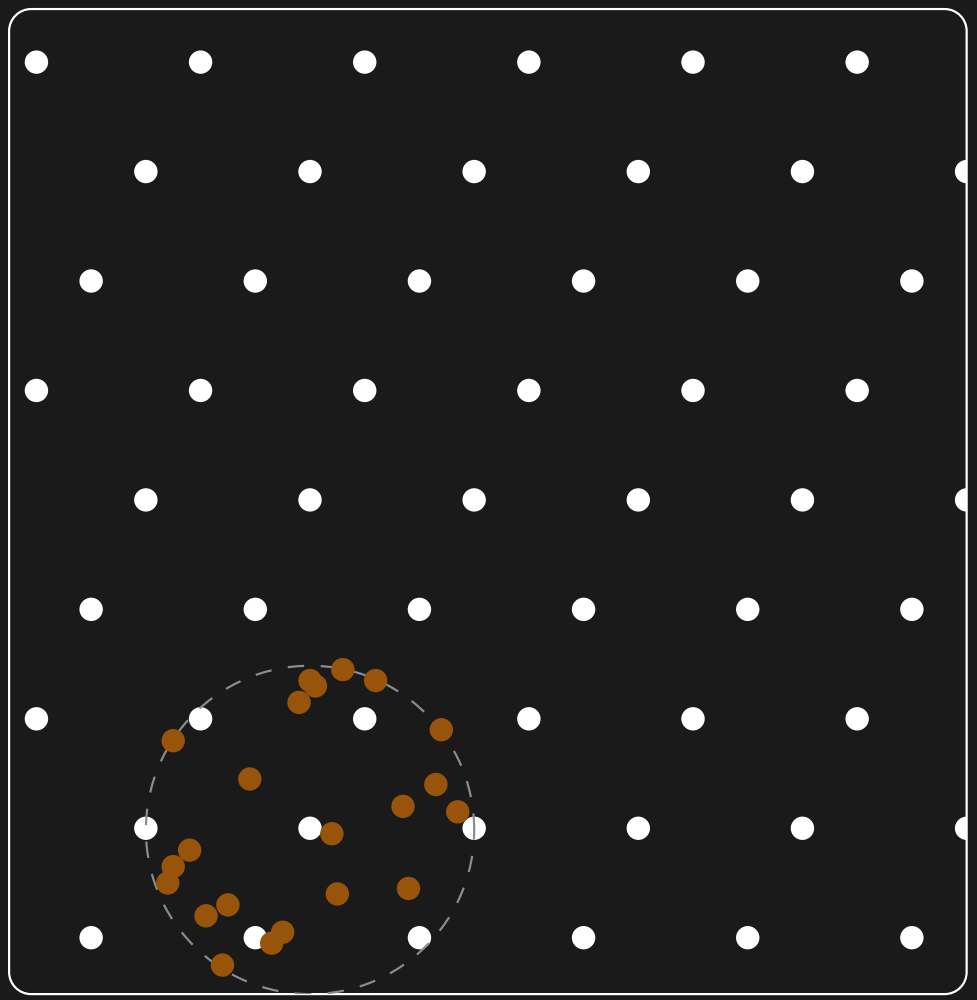
Idea: do not decode deterministically but randomly

Sign$(m, sk)$:

- $x = H(m)$ (hash the message)
- sample $s \in \mathcal{L} \cap \mathcal{B}_r(x)$
  (small radius $r \approx \max_i \|b_i\|$,
  thanks to short basis $b_1, \ldots, b_n$)

[GPV08] Gentry, Peikert, Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC.

# Preventing the attack [GPV08]



Idea: do not decode deterministically but randomly

Sign$(m, sk)$:

- $x = H(m)$ (hash the message)
- sample $s \in \mathcal{L} \cap \mathcal{B}_r(x)$
  (small radius $r \approx \max_i \|b_i\|$,
  thanks to short basis $b_1, \ldots, b_n$)

Lemma: if an adversary can forge signatures, then she can recover a short basis of $\mathcal{L}$ using only $\mathrm{pk}$
(in the ROM)

[GPV08] Gentry, Peikert, Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC.

# Instantiating the framework

Hash-and-sign framework: provably secure if we have a hard lattice $\mathcal{L}$

(i.e., computing a short basis of $\mathcal{L}$ is hard)

# Instantiating the framework

Hash-and-sign framework: provably secure if we have a hard lattice $\mathcal{L}$

(i.e., computing a short basis of $\mathcal{L}$ is hard)

We need: a way to generate random hard lattices

# Instantiating the framework

Hash-and-sign framework: provably secure if we have a hard lattice $\mathcal{L}$

(i.e., computing a short basis of $\mathcal{L}$ is hard)

We need: a way to generate random hard lattices

- we cannot do this provably...
- ... so we make cryptographic assumptions

# Instantiating the framework

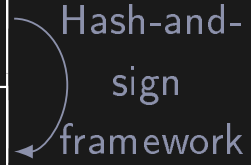
Hash-and-sign framework: provably secure if we have a hard lattice $\mathcal{L}$

(i.e., computing a short basis of $\mathcal{L}$ is hard)

We need: a way to generate random hard lattices

- we cannot do this provably...
- ... so we make cryptographic assumptions

| Assumption | SIS (short integer solution) | | |
|---|---|---|---|
| Construction's name | GPV | | |




[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

[GPV08] Gentry, Peikert, Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. STOC

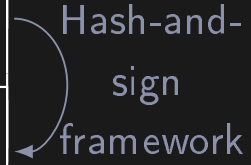
# Instantiating the framework

Hash-and-sign framework: provably secure if we have a hard lattice $\mathcal{L}$
(i.e., computing a short basis of $\mathcal{L}$ is hard)

We need: a way to generate random hard lattices

- we cannot do this provably...
- ... so we make cryptographic assumptions

| Assumption | SIS (short integer solution) | NTRU | |
|---|---|---|---|
| Construction's name | GPV | Falcon | |




[HPS98] Hoffstein, Pipher, and Silverman. NTRU: a ring based public key cryptosystem. ANTS.

[Falcon] Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Prest, Ricosset, Seiler, Whyte, Zhang. NIST standard

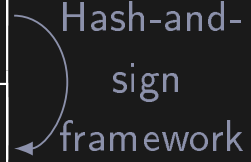
# Instantiating the framework

Hash-and-sign framework: provably secure if we have a hard lattice $\mathcal{L}$
(i.e., computing a short basis of $\mathcal{L}$ is hard)

We need: a way to generate random hard lattices

- we cannot do this provably...
- ... so we make cryptographic assumptions

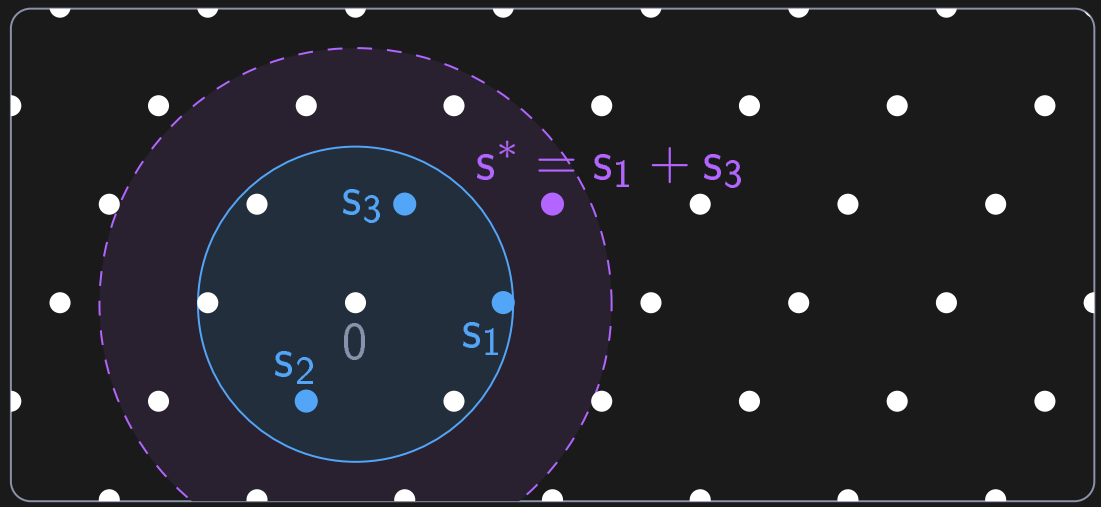
| Assumption | SIS (short integer solution) | NTRU | LIP (lattice isomorphism problem) |
|---|---|---|---|
| Construction's name | GPV | Falcon | Hawk |




[DW22] Ducas and van Woerden. On the lattice isomorphism problem, quadratic forms [...]. Eurocrypt

[DPPW23] Ducas, Postlethwaite, Pulles, van Woerden. Hawk: Module LIP makes lattice signatures [...]. Asiacrypt

# Attack on a knowledge variant of SIS



# Short Integer Solution [Ajt96]

## Short Integer Solution (SIS) problem:

**Parameters:** $m \geq n$, $q \gg B$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

# Short Integer Solution [Ajt96]

## Short Integer Solution (SIS) problem:

**Parameters:** $m \geq n$, $q \gg B$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

# Short Integer Solution [Ajt96]

## Short Integer Solution (SIS) problem:

**Parameters:** $m \geq n$, $q \gg B$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

Output: $s \in \mathbb{Z}^m$ short ($\|s\| \leq B$)

s.t. $s \cdot A = 0 \bmod q$

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

# Short Integer Solution [Ajt96]

## Short Integer Solution (SIS) problem:

**Parameters:** $m \geq n$, $q \gg B$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

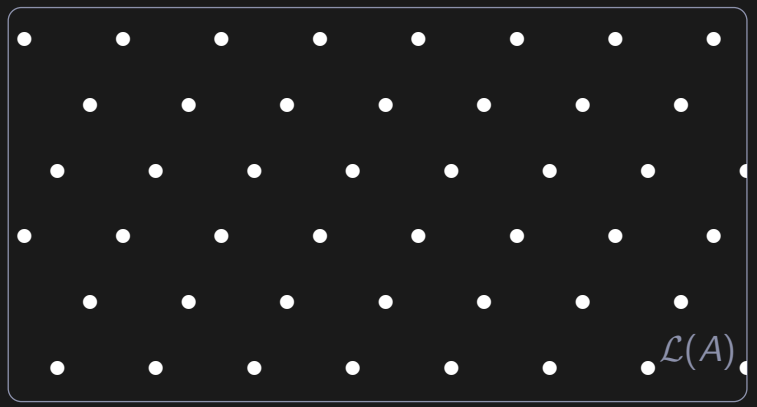
Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

Output: $s \in \mathbb{Z}^m$ short ($\|s\| \leq B$)

s.t. $s \, A = 0 \bmod q$

$$\mathcal{L}(A) := \{v \in \mathbb{Z}^m \mid vA = 0 \bmod q\}$$

is a lattice




[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

# Short Integer Solution [Ajt96]

## Short Integer Solution (SIS) problem:

**Parameters:** $m \geq n$, $q \gg B$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

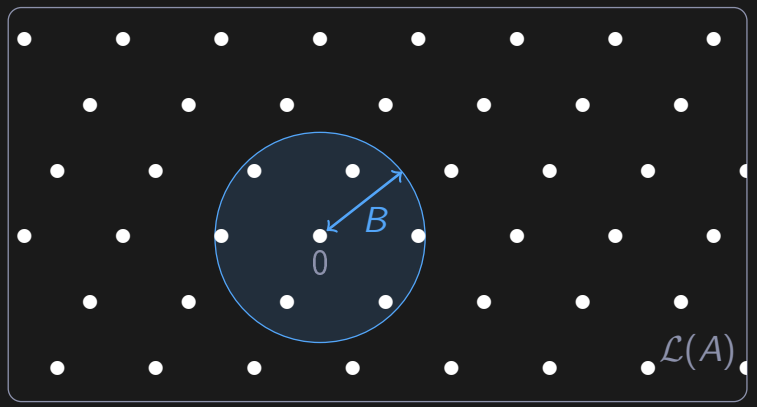
Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

Output: $s \in \mathbb{Z}^m$ short ($\|s\| \leq B$)

s.t. $s \; A = 0 \bmod q$

$$\mathcal{L}(A) := \{v \in \mathbb{Z}^m \mid vA = 0 \bmod q\}$$

is a lattice




SIS = find short vector in $\mathcal{L}(A)$

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

# Short Integer Solution [Ajt96]

## Short Integer Solution (SIS) problem:

**Parameters:** $m \geq n$, $q \gg B$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

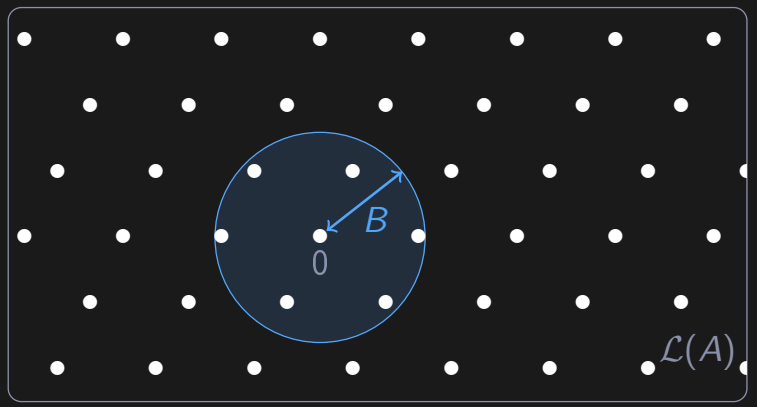
Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

Output: $s \in \mathbb{Z}^m$ short ($\|s\| \leq B$)

s.t. $s \; A = 0 \bmod q$

$$\mathcal{L}(A) := \{v \in \mathbb{Z}^m \mid vA = 0 \bmod q\}$$

is a lattice




SIS = find short vector in $\mathcal{L}(A)$

## SIS assumption

Solving the SIS problem is hard

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

## knowledge SIS problem:

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

[ACLMT22] Albrecht, Cini, Lai, Malavolta,Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

knowledge SIS problem:

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

$+ s_1, \ldots, s_k \in \mathbb{Z}^m$ short, $s_i A = 0 \bmod q$

[ACLMT22] Albrecht, Cini, Lai, Malavolta, Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

## knowledge SIS problem:

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

$+ s_1, \ldots, s_k \in \mathbb{Z}^m$ short, $s_i A = 0 \bmod q$

Output: $s^* \in \mathbb{Z}^m$ somewhat short

s.t. $s^* A = 0 \bmod q$

[ACLMT22] Albrecht, Cini, Lai, Malavolta, Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

## knowledge SIS problem:

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

$+ s_1, \ldots, s_k \in \mathbb{Z}^m$ short, $s_i A = 0 \bmod q$

Output: $s^* \in \mathbb{Z}^m$ somewhat short

s.t. $s^* A = 0 \bmod q$

## knowledge SIS assumption

The only way to find such $s^*$ is by taking a small combination of the $s_i$'s


[ACLMT22] Albrecht, Cini, Lai, Malavolta, Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

## knowledge SIS problem:

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

$+ s_1, \ldots, s_k \in \mathbb{Z}^m$ short, $s_i A = 0 \bmod q$

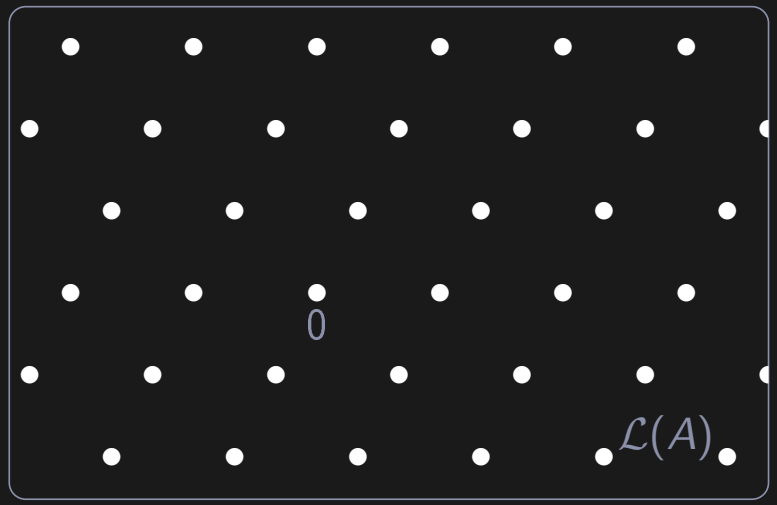
Output: $s^* \in \mathbb{Z}^m$ somewhat short

s.t. $s^* A = 0 \bmod q$

## knowledge SIS assumption

The only way to find such $s^*$ is by taking a small combination of the $s_i$'s

$\mathcal{L}(A) := \{v \in \mathbb{Z}^m \mid vA = 0 \bmod q\}$





[ACLMT22] Albrecht, Cini, Lai, Malavolta, Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

## knowledge SIS problem:

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

$+ s_1, \ldots, s_k \in \mathbb{Z}^m$ short, $s_i A = 0 \bmod q$

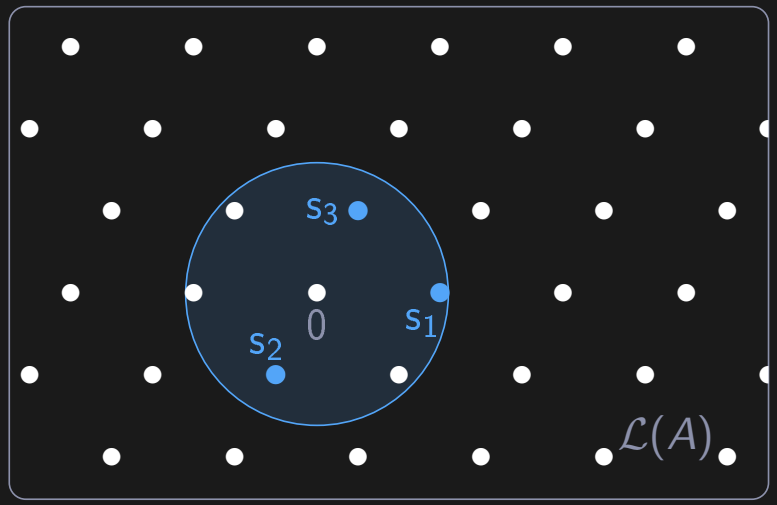
Output: $s^* \in \mathbb{Z}^m$ somewhat short

s.t. $s^* A = 0 \bmod q$

## knowledge SIS assumption

The only way to find such $s^*$ is by taking a small combination of the $s_i$'s

$\mathcal{L}(A) := \{v \in \mathbb{Z}^m \mid vA = 0 \bmod q\}$





[ACLMT22] Albrecht, Cini, Lai, Malavolta, Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

**knowledge SIS problem:**

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

**Input:** $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

$+ s_1, \ldots, s_k \in \mathbb{Z}^m$ short, $s_i A = 0 \bmod q$

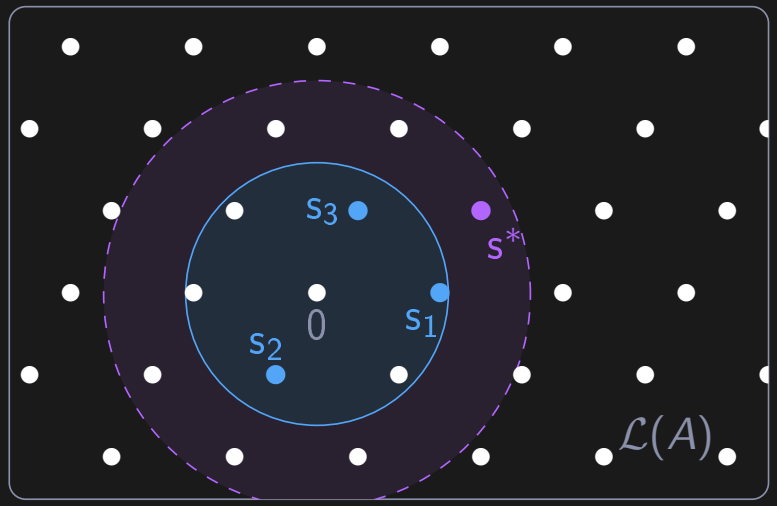
**Output:** $s^* \in \mathbb{Z}^m$ somewhat short

s.t. $s^* A = 0 \bmod q$

**knowledge SIS assumption**

The only way to find such $s^*$ is by taking a small combination of the $s_i$'s

$\mathcal{L}(A) := \{v \in \mathbb{Z}^m \mid vA = 0 \bmod q\}$





[ACLMT22] Albrecht, Cini, Lai, Malavolta, Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Knowledge variant of SIS (adapted from [ACLMT22, Alb22])

**knowledge SIS problem:**

**Parameters:** $k \geq m \geq n$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

**Input:** $A \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$

$+ s_1, \ldots, s_k \in \mathbb{Z}^m$ short, $s_i A = 0 \bmod q$

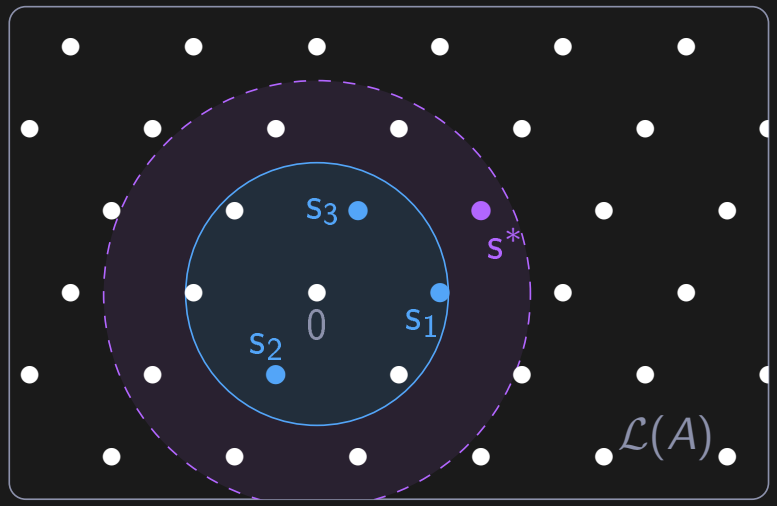
**Output:** $s^* \in \mathbb{Z}^m$ somewhat short

s.t. $s^* A = 0 \bmod q$

## knowledge SIS assumption

The only way to find such $s^*$ is by taking a small combination of the $s_i$'s

$\mathcal{L}(A) := \{v \in \mathbb{Z}^m \mid vA = 0 \bmod q\}$




## knowledge SIS (geometric)

Given $k$ short vectors of $\mathcal{L}(A)$, the only way to produce a new short vector is by taking a small combination of these $k$ vectors.


[ACLMT22] Albrecht, Cini, Lai, Malavolta, Thyagarajan. Lattice-based SNARKs [...]. Crypto

[Alb22] Albrecht's blogpost: The k-R-ISIS (of knowledge) assumption

# Summary

## knowledge SIS assumption

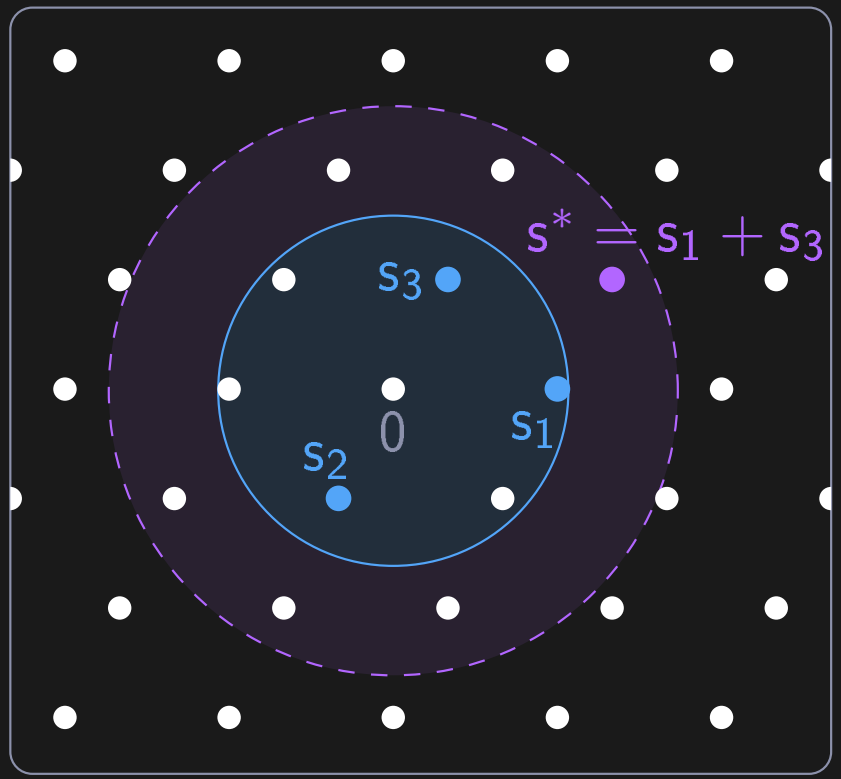
(geometric formalism)

**Parameters:** $k \geq n \geq 0$, $B > 0$

Input:

- $\mathcal{L}$ lattice of dim $n$
- $s_1, \ldots, s_k \in \mathcal{L}$ short ($\|s_i\| \leq B$)

Assumption: the only way to compute new somewhat short ($\leq \mathrm{poly}(n) \cdot B$) vectors in $\mathcal{L}$ is by taking small combinations of the $s_i$'s.



# Summary

## knowledge SIS assumption

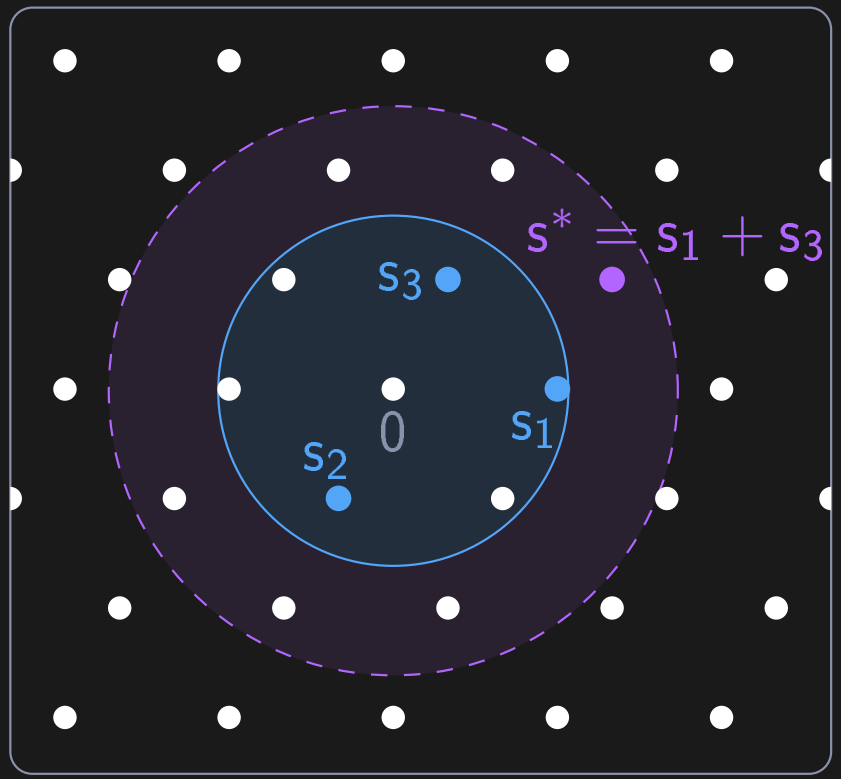
(geometric formalism)

**Parameters:** $k \geq n \geq 0$, $B > 0$

Input:

- $\mathcal{L}$ lattice of dim $n$
- $s_1, \ldots, s_k \in \mathcal{L}$ short ($\|s_i\| \leq B$)

Assumption: the only way to compute new somewhat short ($\leq \mathrm{poly}(n) \cdot B$) vectors in $\mathcal{L}$ is by taking small combinations of the $s_i$'s.



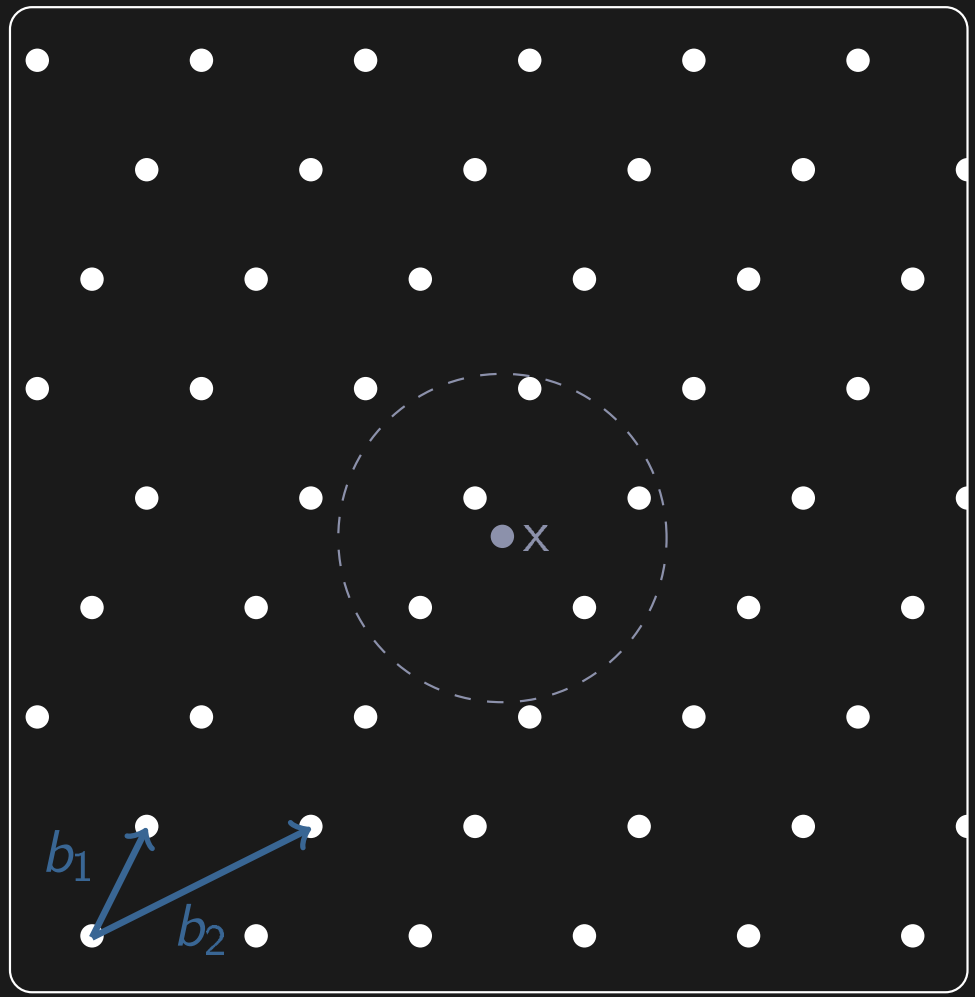

Really?

# Tool: Sampling in a lattice [PP21]

Input: center x, radius $r$

(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$




[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Tool: Sampling in a lattice [PP21]

Input: center x, radius $r$

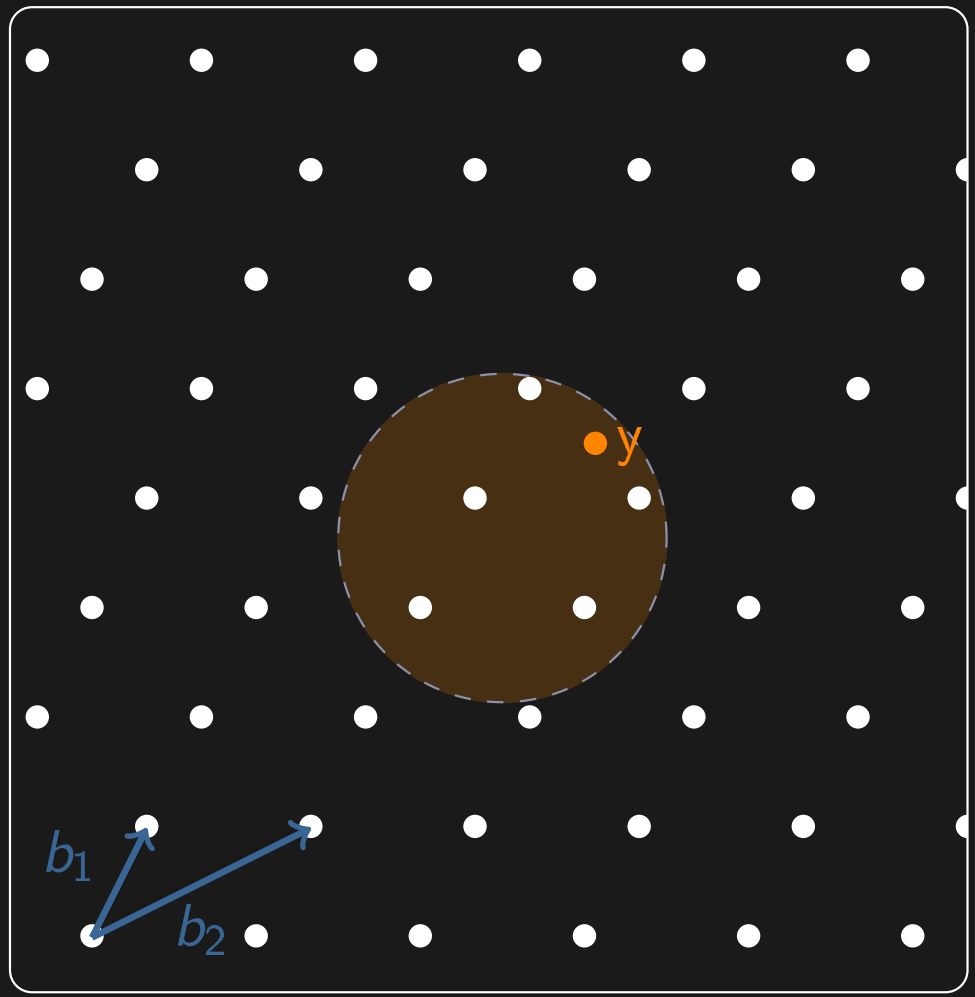
(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$

Algo:

- Sample $y \leftarrow \mathcal{U}(\mathcal{B}_r(x))$

  (continuous distribution)




[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Tool: Sampling in a lattice [PP21]

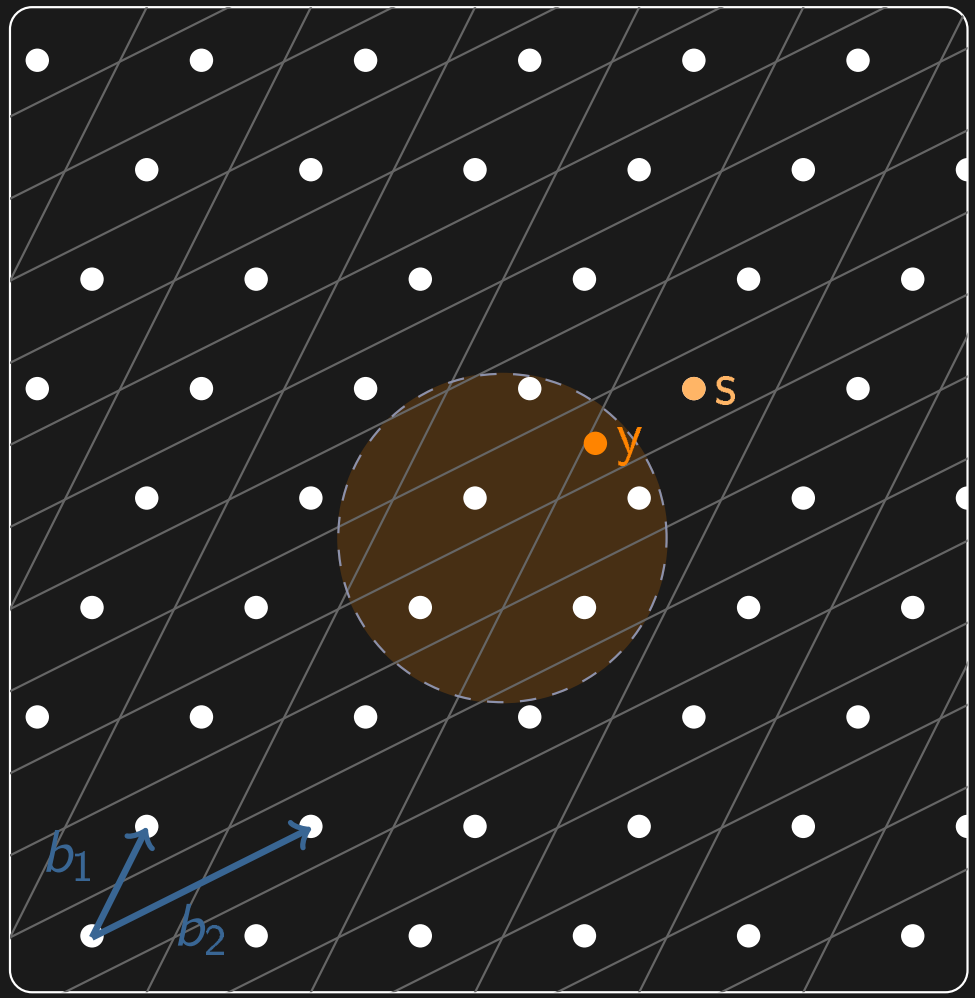
Input: center x, radius $r$

(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$

Algo:

- Sample $y \leftarrow \mathcal{U}(\mathcal{B}_r(x))$
  (continuous distribution)
- $s \leftarrow \texttt{Babai\_decoding}(y)$




[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Tool: Sampling in a lattice [PP21]

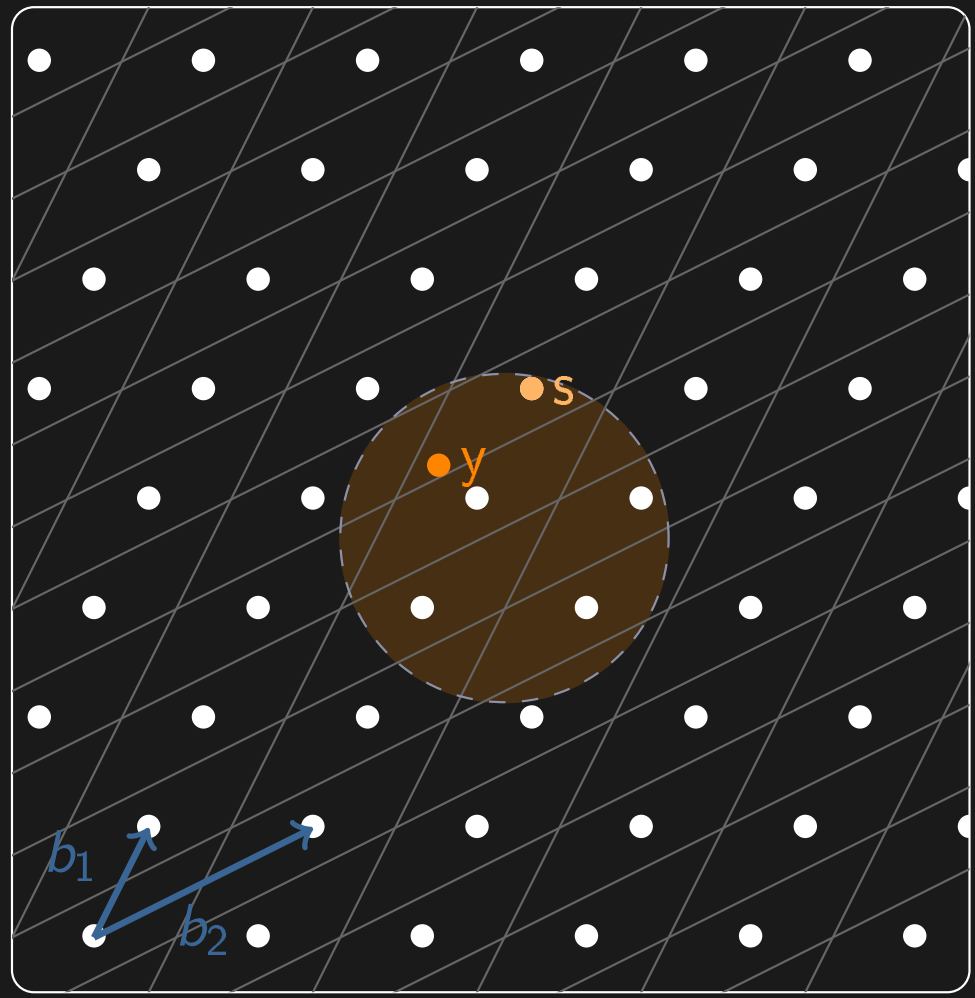
Input: center x, radius $r$

(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$

Algo:

- Sample $y \leftarrow \mathcal{U}(\mathcal{B}_r(x))$
  (continuous distribution)
- $s \leftarrow \texttt{Babai\_decoding}(y)$
- repeat until $s \in \mathcal{B}_r(x)$




[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Tool: Sampling in a lattice [PP21]

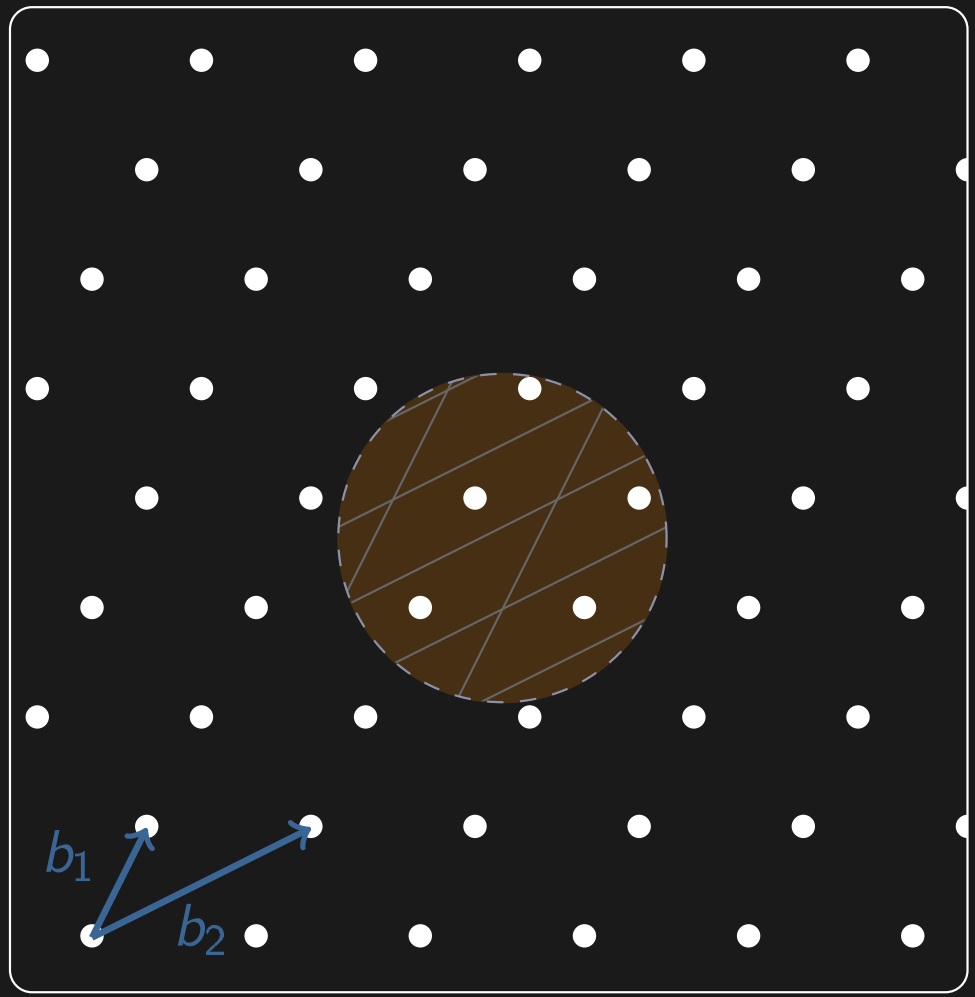
Input: center x, radius $r$

(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$

Algo:

- Sample $y \leftarrow \mathcal{U}(\mathcal{B}_r(x))$
  (continuous distribution)
- $s \leftarrow \texttt{Babai\_decoding}(y)$
- repeat until $s \in \mathcal{B}_r(x)$




[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Tool: Sampling in a lattice [PP21]

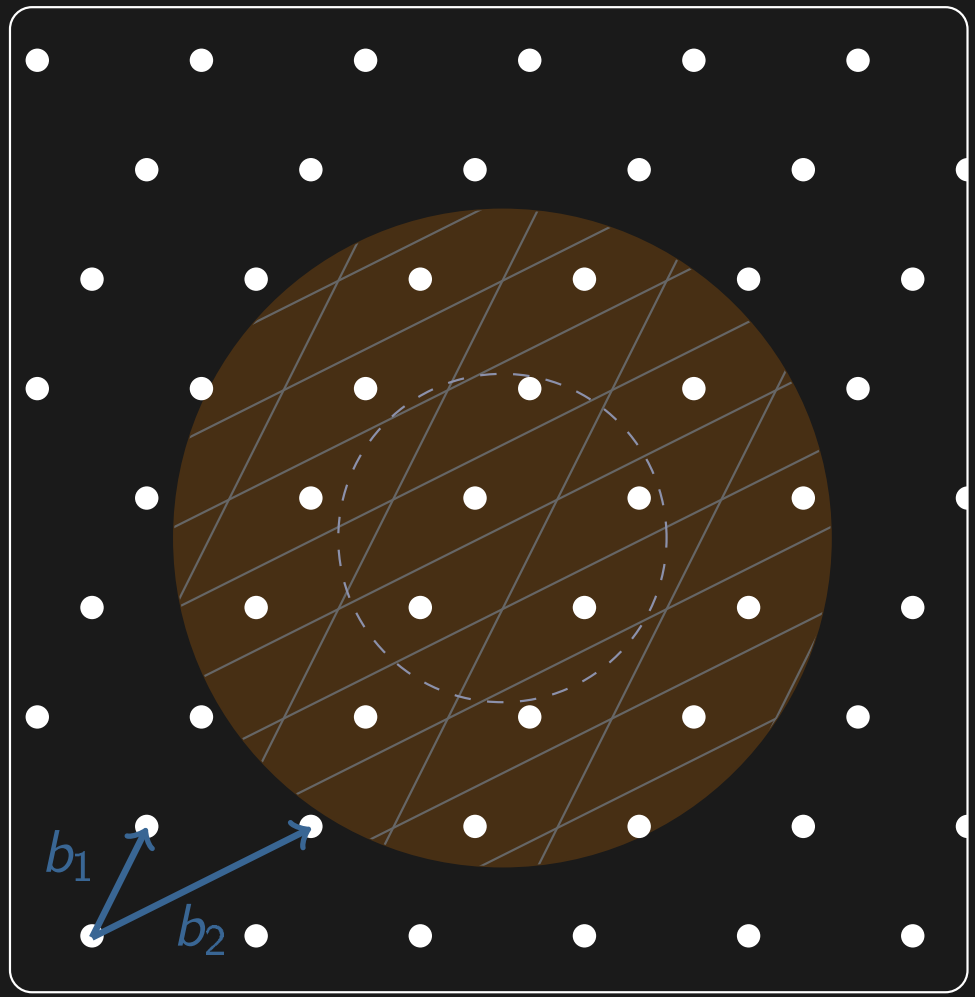
Input: center x, radius $r$

(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$

Algo:

- Sample $y \leftarrow \mathcal{U}(\mathcal{B}_{r'}(x))$
  (continuous distribution)
- $s \leftarrow \texttt{Babai\_decoding}(y)$
- repeat until $s \in \mathcal{B}_r(x)$




[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Tool: Sampling in a lattice [PP21]

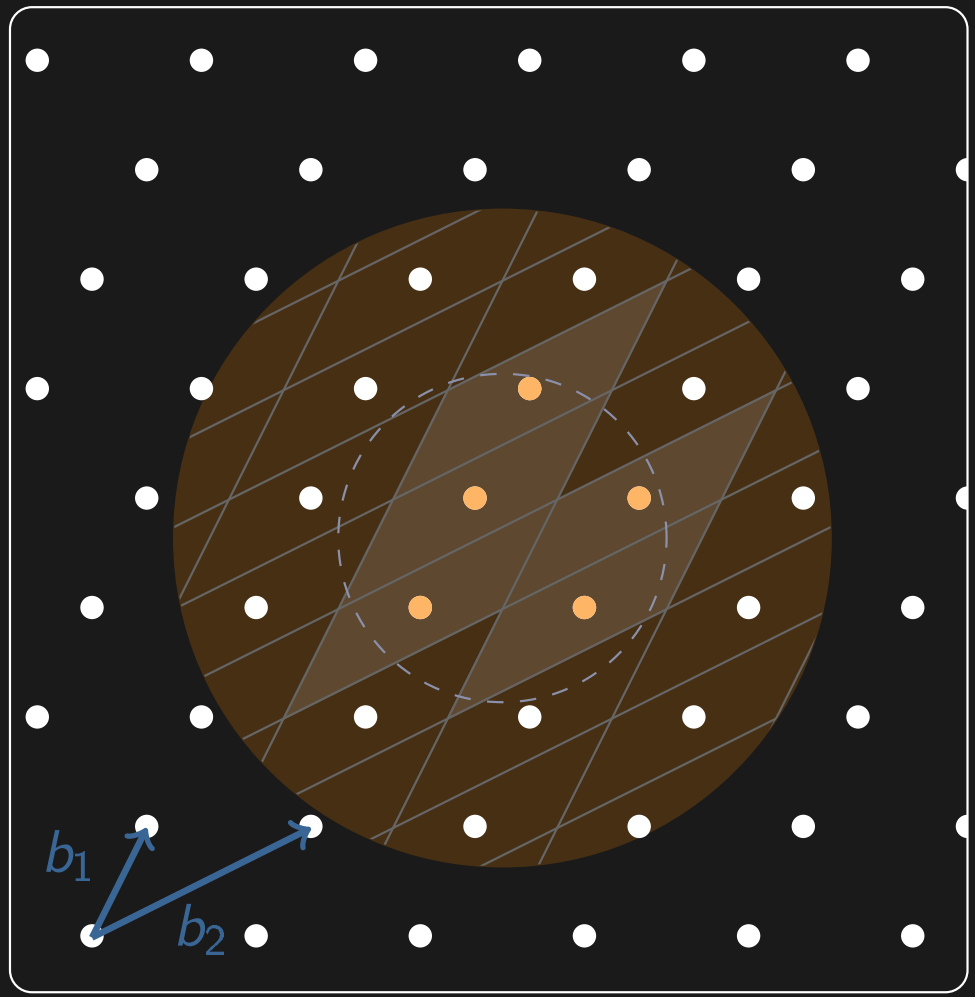
Input: center x, radius $r$

(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$

Algo:

- Sample $y \leftarrow \mathcal{U}(\mathcal{B}_{r'}(x))$
  (continuous distribution)
- $s \leftarrow \texttt{Babai\_decoding}(y)$
- repeat until $s \in \mathcal{B}_r(x)$




[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Tool: Sampling in a lattice [PP21]

Input: center x, radius $r$

(and a short basis $(b_1, \ldots, b_n)$ of $\mathcal{L}$)

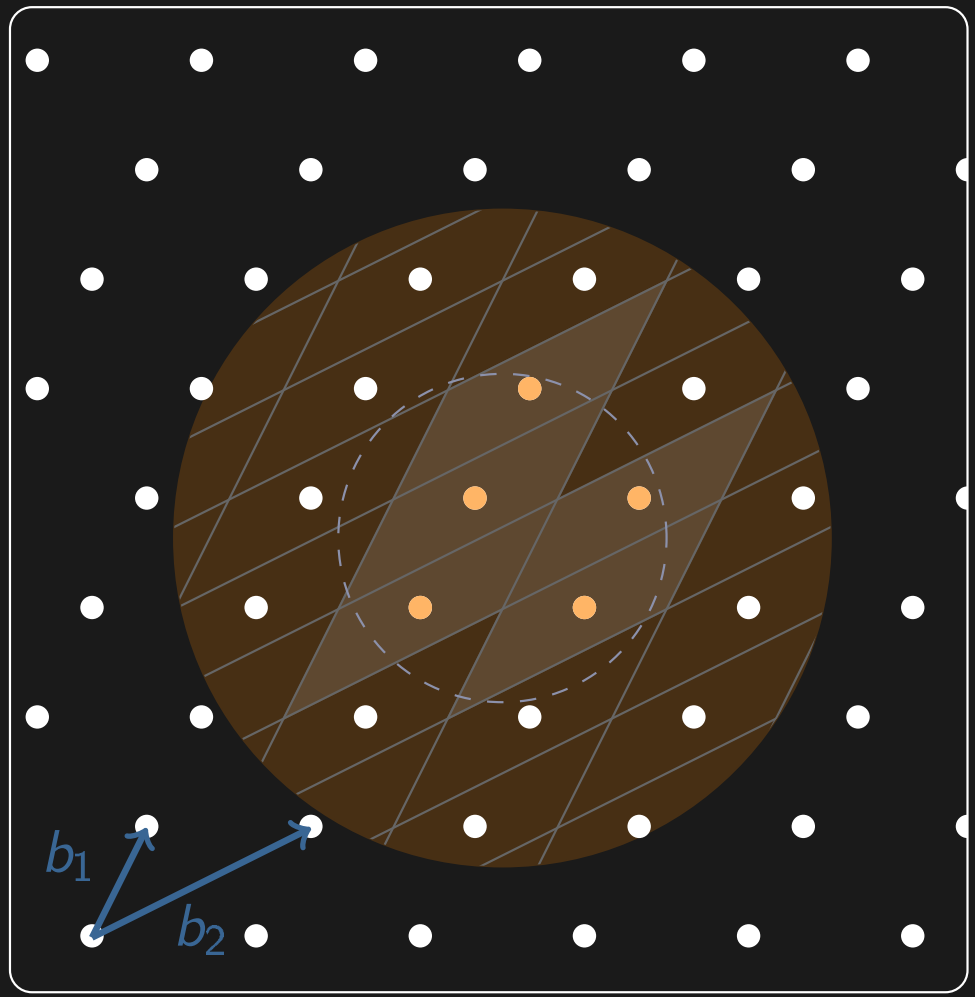
Output: $s \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r(x))$

Algo:

- Sample $y \leftarrow \mathcal{U}(\mathcal{B}_{r'}(x))$
  (continuous distribution)
- $s \leftarrow \texttt{Babai\_decoding}(y)$
- repeat until $s \in \mathcal{B}_r(x)$

polynomial time if

$r \geq 2n^2 \cdot \max_i \|b_i\|$

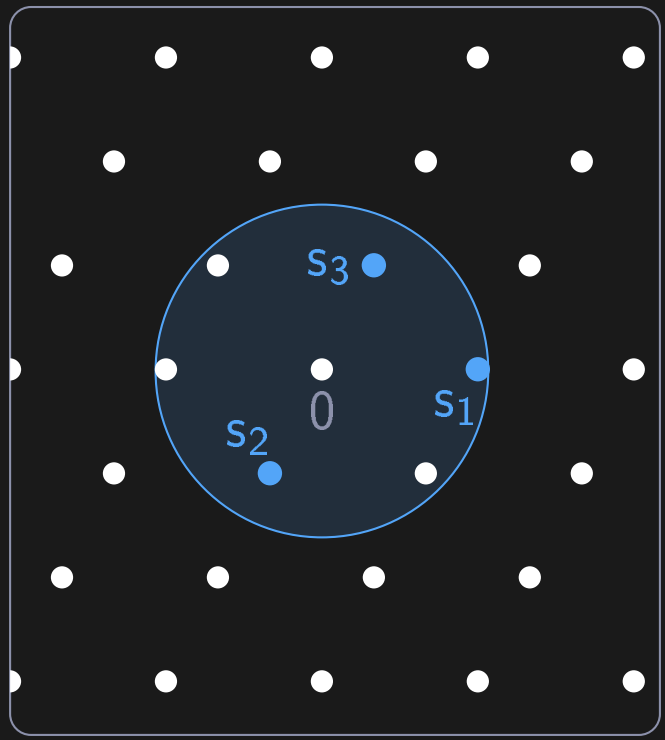



[PP21] Plançon and Prest. Exact Lattice Sampling from Non-Gaussian Distributions. PKC.

# Contradicting knowledge SIS (adapted from [WW23])

## Reminder: knowledge SIS

Given $s_1, \ldots, s_k \in \mathcal{L}$ short ($\mathcal{L}$ dim $n$, $k \geq n$)
the only way to find $s^* \in \mathcal{L}$ short, is by taking small combinations of the $s_i$'s




[WW23] Wee and Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. Asiacrypt

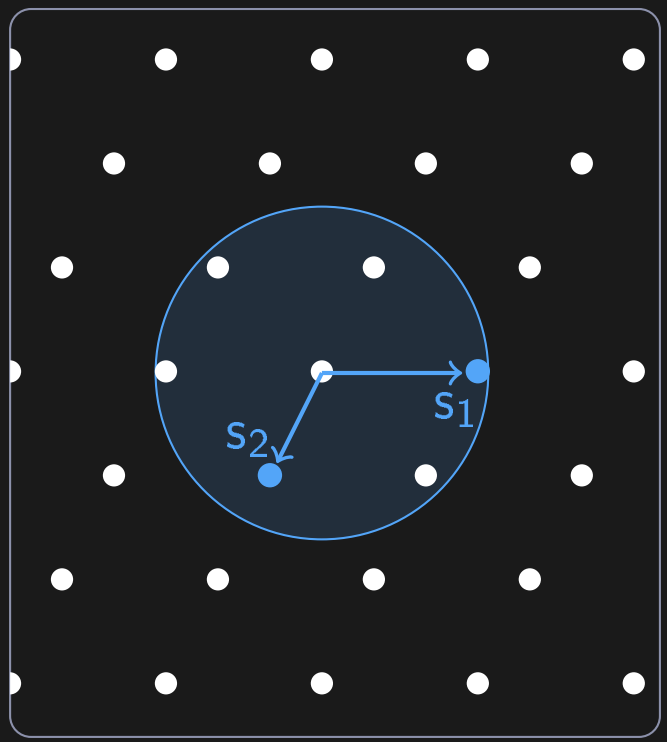
# Contradicting knowledge SIS (adapted from [WW23])

## Reminder: knowledge SIS

Given $s_1, \ldots, s_k \in \mathcal{L}$ short ($\mathcal{L}$ dim $n$, $k \geq n$)
the only way to find $s^* \in \mathcal{L}$ short, is by taking small combinations of the $s_i$'s

Contradiction algorithm:

1. assume $s_1, \ldots, s_n$ basis of $\mathcal{L}$
   (we can make them a basis if they are not)




[WW23] Wee and Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. Asiacrypt

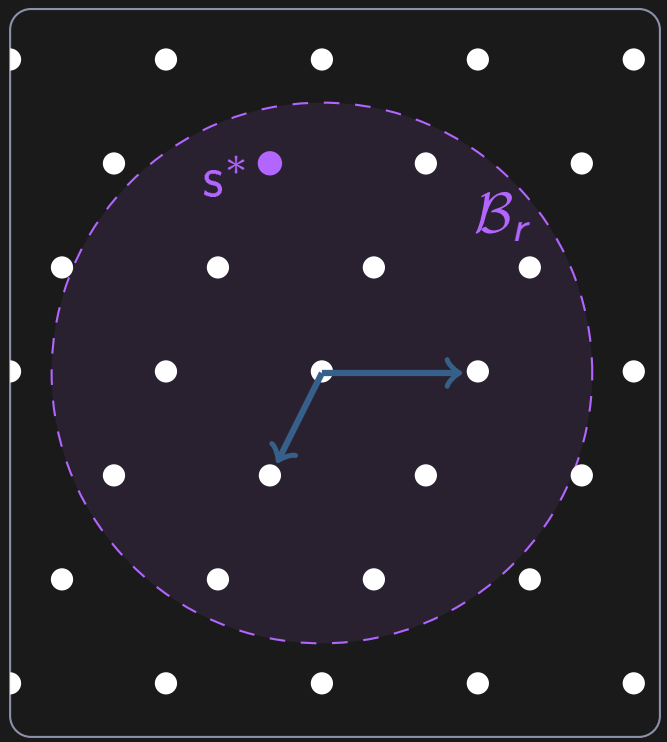
# Contradicting knowledge SIS (adapted from [WW23])

## Reminder: knowledge SIS

Given $s_1, \ldots, s_k \in \mathcal{L}$ short ($\mathcal{L}$ dim $n$, $k \geq n$)
the only way to find $s^* \in \mathcal{L}$ short, is by taking small combinations of the $s_i$'s

Contradiction algorithm:

1. assume $s_1, \ldots, s_n$ basis of $\mathcal{L}$
   (we can make them a basis if they are not)
2. sample $s^* \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r)$ (using $s_1, \ldots, s_n$)
   ($r = 2n^2 \cdot \max \|s_i\|$)




[WW23] Wee and Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. Asiacrypt

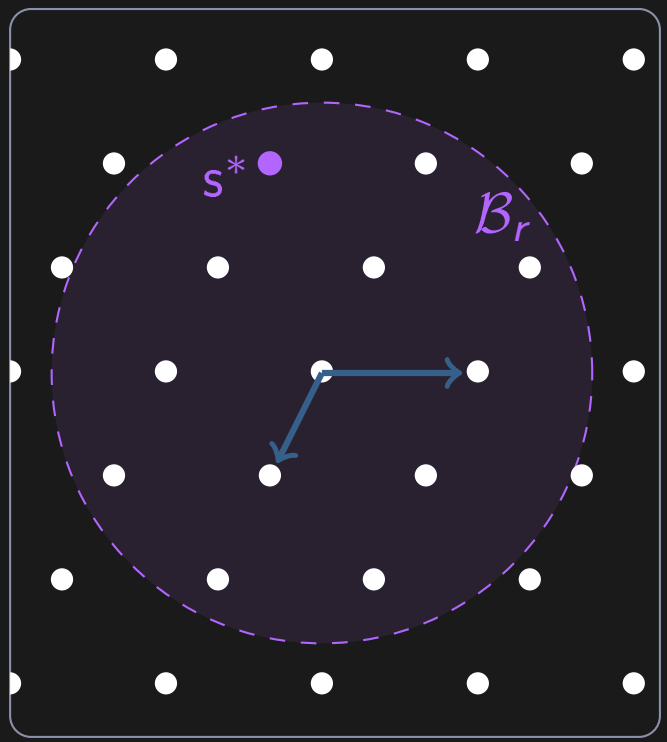
# Contradicting knowledge SIS (adapted from [WW23])

## Reminder: knowledge SIS

Given $s_1, \ldots, s_k \in \mathcal{L}$ short ($\mathcal{L}$ dim $n$, $k \geq n$)
the only way to find $s^* \in \mathcal{L}$ short, is by taking small combinations of the $s_i$'s

### Contradiction algorithm:

1. assume $s_1, \ldots, s_n$ basis of $\mathcal{L}$
   (we can make them a basis if they are not)
2. sample $s^* \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r)$ (using $s_1, \ldots, s_n$)
   ($r = 2n^2 \cdot \max \|s_i\|$)




### Properties

$\rightsquigarrow$ $s^* \in \mathcal{L}$ is somewhat short

[WW23] Wee and Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. Asiacrypt

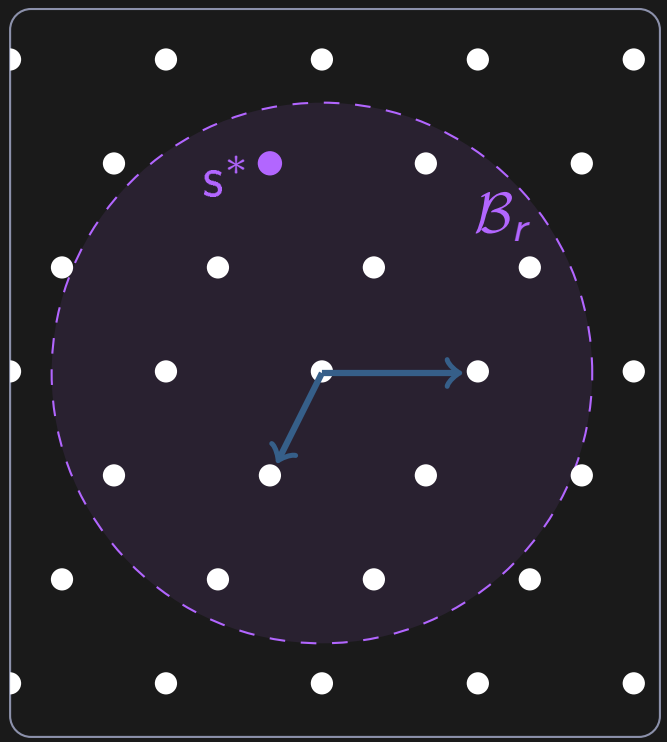
# Contradicting knowledge SIS (adapted from [WW23])

## Reminder: knowledge SIS

Given $s_1, \ldots, s_k \in \mathcal{L}$ short ($\mathcal{L}$ dim $n$, $k \geq n$)
the only way to find $s^* \in \mathcal{L}$ short, is by taking small combinations of the $s_i$'s

### Contradiction algorithm:

1. assume $s_1, \ldots, s_n$ basis of $\mathcal{L}$
   (we can make them a basis if they are not)
2. sample $s^* \leftarrow \mathcal{U}(\mathcal{L} \cap \mathcal{B}_r)$ (using $s_1, \ldots, s_n$)
   ($r = 2n^2 \cdot \max \|s_i\|$)

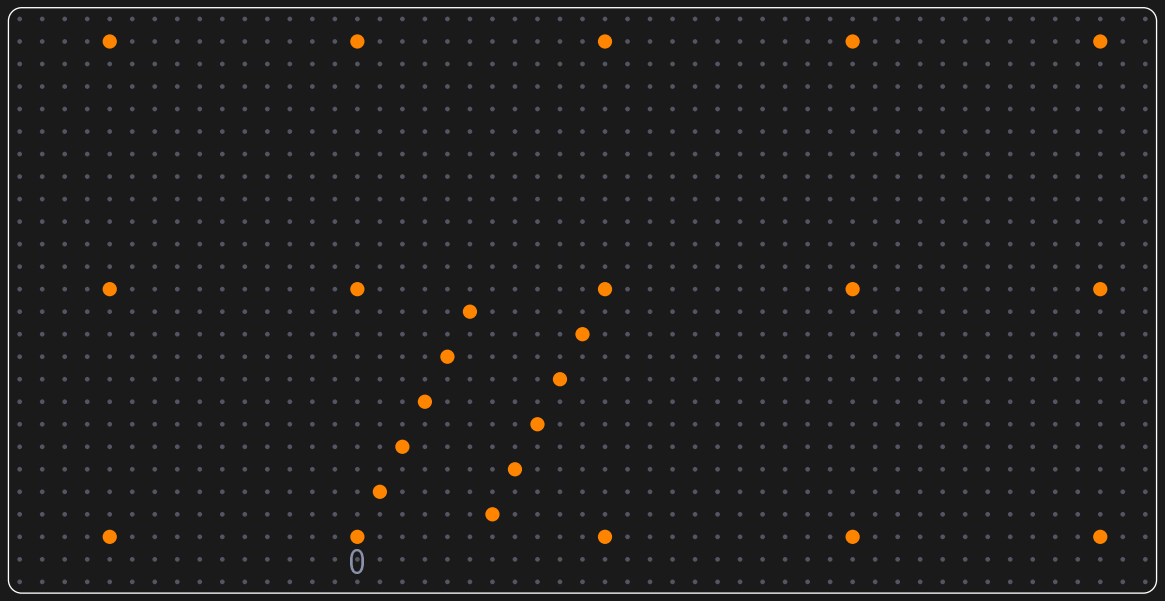



### Properties

- $\rightsquigarrow$ $s^* \in \mathcal{L}$ is somewhat short
- $\rightsquigarrow$ $s^*$ is not a small combination of $s_1, \ldots, s_n$ with high probability
  (it might be a small combination of $s_1, \ldots, s_k$)

[WW23] Wee and Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. Asiacrypt

# The NTRU problem



# NTRU, matrix formalism [HPS98]

## NTRU problem

**Parameters:** $q$ prime, $B \ll \sqrt{q}$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $h \in \mathbb{Z}_q$

Output: $f, g \in \mathbb{Z}$ such that $0 < |f|, |g| \leq B$ and

$$h = f \cdot g^{-1} \bmod q \quad (\Leftrightarrow gh = f \bmod q)$$

[HPS98] Hoffstein, Pipher, and Silverman. NTRU: a ring based public key cryptosystem. ANTS.

# NTRU, matrix formalism [HPS98]

## NTRU problem

**Parameters:** $q$ prime, $B \ll \sqrt{q}$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $h \in \mathbb{Z}_q$

Output: $f, g \in \mathbb{Z}$ such that $0 < |f|, |g| \leq B$ and

$$h = f \cdot g^{-1} \bmod q \quad (\Leftrightarrow gh = f \bmod q)$$

Remark: for most $h \in \mathbb{Z}_q$, there is no solution

$$|\mathbb{Z}_q| = q, \quad \left|\left\{(f,g) \,\middle|\, 0 < |f|, |g| \leq B\right\}\right| = (2B)^2$$

[HPS98] Hoffstein, Pipher, and Silverman. NTRU: a ring based public key cryptosystem. ANTS.

# NTRU, matrix formalism [HPS98]

## NTRU problem

**Parameters:** $q$ prime, $B \ll \sqrt{q}$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $h \in \mathbb{Z}_q$

Output: $f, g \in \mathbb{Z}$ such that $0 < |f|, |g| \leq B$ and

$$h = f \cdot g^{-1} \bmod q \quad (\Leftrightarrow gh = f \bmod q)$$

Remark: for most $h \in \mathbb{Z}_q$, there is no solution

$$|\mathbb{Z}_q| = q \gg \left|\left\{(f,g) \,\middle|\, 0 < |f|, |g| \leq B\right\}\right| = (2B)^2$$

[HPS98] Hoffstein, Pipher, and Silverman. NTRU: a ring based public key cryptosystem. ANTS.

# NTRU, matrix formalism [HPS98]

## NTRU problem

**Parameters:** $q$ prime, $B \ll \sqrt{q}$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $h \in \mathbb{Z}_q$ for which a solution exists

Output: $f, g \in \mathbb{Z}$ such that $0 < |f|, |g| \leq B$ and

$$h = f \cdot g^{-1} \bmod q \quad (\Leftrightarrow gh = f \bmod q)$$

Remark: for most $h \in \mathbb{Z}_q$, there is no solution

$$|\mathbb{Z}_q| = q \gg \left|\left\{(f,g) \,\middle|\, 0 < |f|, |g| \leq B\right\}\right| = (2B)^2$$

[HPS98] Hoffstein, Pipher, and Silverman. NTRU: a ring based public key cryptosystem. ANTS.

# NTRU, matrix formalism [HPS98]

## NTRU problem

**Parameters:** $q$ prime, $B \ll \sqrt{q}$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

Input: $h \in \mathbb{Z}_q$ for which a solution exists

Output: $f, g \in \mathbb{Z}$ such that $0 < |f|, |g| \leq B$ and

$$h = f \cdot g^{-1} \bmod q \quad (\Leftrightarrow gh = f \bmod q)$$

Remark: for most $h \in \mathbb{Z}_q$, there is no solution

$$|\mathbb{Z}_q| = q \gg \left|\left\{(f,g) \,\middle|\, 0 < |f|, |g| \leq B\right\}\right| = (2B)^2$$

**NTRU assumption**

Solving the NTRU problem is hard

---

[HPS98] Hoffstein, Pipher, and Silverman. NTRU: a ring based public key cryptosystem. ANTS.

# NTRU, geometric formalism

NTRU problem (matrix formalism): given $h \in \mathbb{Z}_q$ (for which a solution exists), find $(f, g)$ with $0 < |f|, |g| \leq B$ and $h = f \cdot g^{-1} \bmod q$

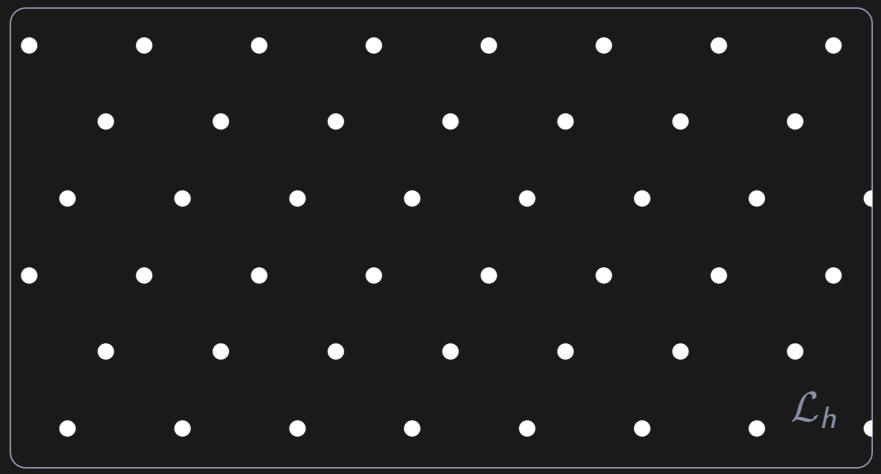
# NTRU, geometric formalism

NTRU problem (matrix formalism): given $h \in \mathbb{Z}_q$ (for which a solution exists), find $(f, g)$ with $0 < |f|, |g| \leq B$ and $h = f \cdot g^{-1} \bmod q$

## NTRU lattice

$$\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$$

(lattice of dimension 2)



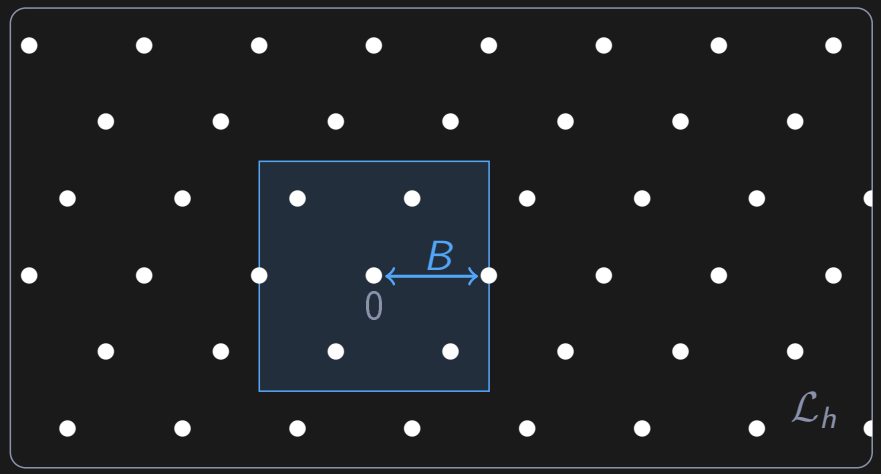
# NTRU, geometric formalism

NTRU problem (matrix formalism): given $h \in \mathbb{Z}_q$ (for which a solution exists), find $(f, g)$ with $0 < |f|, |g| \leq B$ and $h = f \cdot g^{-1} \bmod q$

## NTRU lattice

$$\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$$

(lattice of dimension 2)




$(f, g)$ solution for NTRU $\iff$ $(f, g) \in \mathcal{L}_h \setminus \{0\}$ and $\|(f, g)\|_\infty \leq B$

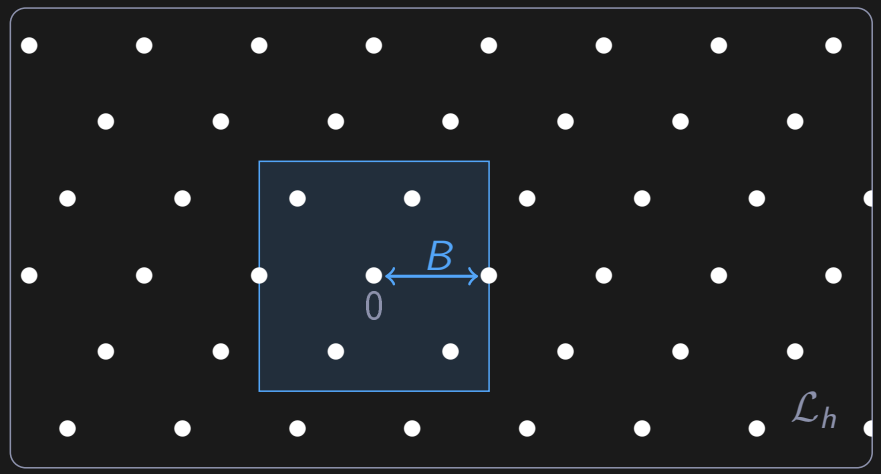
# NTRU, geometric formalism

NTRU problem (matrix formalism): given $h \in \mathbb{Z}_q$ (for which a solution exists), find $(f, g)$ with $0 < |f|, |g| \leq B$ and $h = f \cdot g^{-1} \bmod q$

## NTRU lattice

$$\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$$

(lattice of dimension 2)




$(f, g)$ solution for NTRU $\iff$ $(f, g) \in \mathcal{L}_h \setminus \{0\}$ and $\|(f, g)\|_\infty \leq B$

solving NTRU $\iff$ finding a short non-zero vector in $\mathcal{L}_h$

# Summing up

**Reminder:** SVP = short vector problem

We have seen


(if you can solve SVP, you can also solve NTRU)

# Summing up

**Reminder:** SVP = short vector problem

We have seen


(if you can solve SVP, you can also solve NTRU)

How about the other direction? ...


(if you can solve NTRU, can you solve SVP?)

# Summing up

**Reminder:** SVP = short vector problem

We have seen


(if you can solve SVP, you can also solve NTRU)

How about the other direction? . . .


(if you can solve NTRU, can you solve SVP?)

. . . we don't know (but it seems unlikely)

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$


$\mathcal{L}_h$ contains

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$


$\mathcal{L}_h$ contains

$(q, 0)$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$


$\mathcal{L}_h$ contains

$(q, 0), \quad (0, q)$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$


$\mathcal{L}_h$ contains

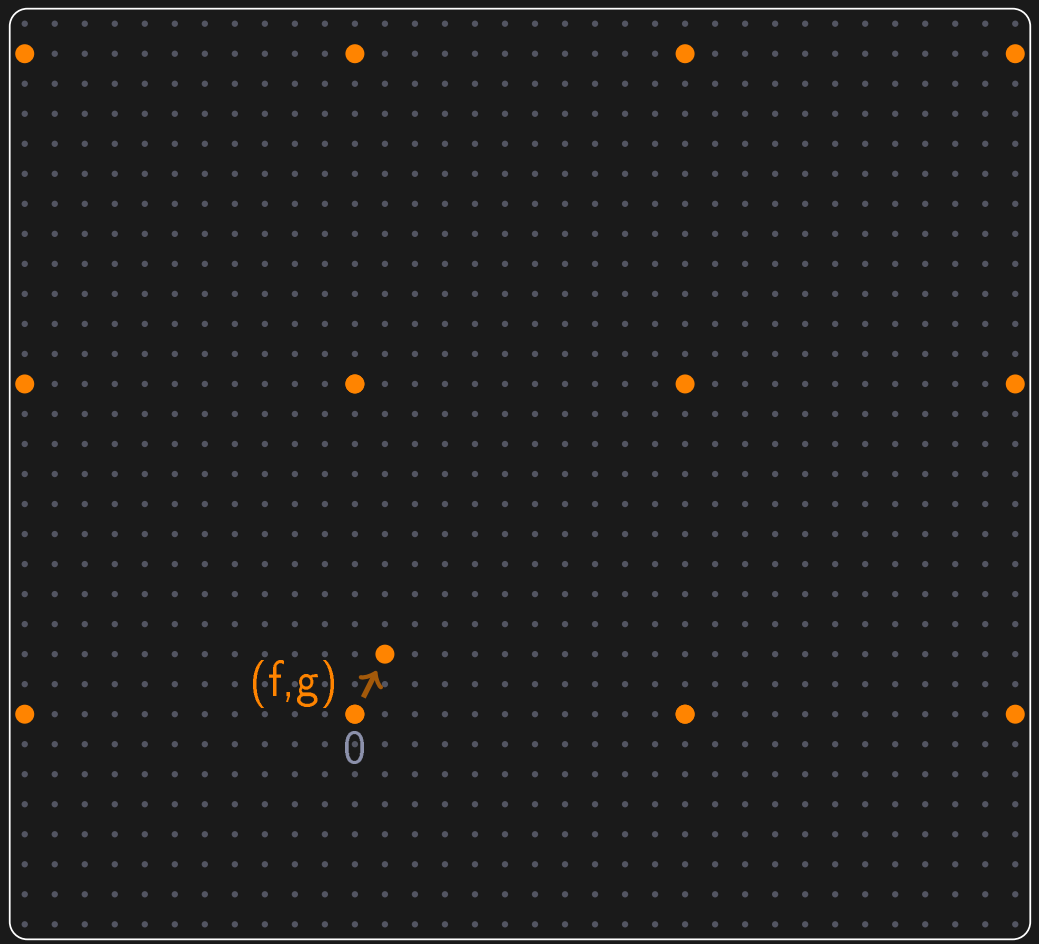
$\underbrace{(q, 0), \quad (0, q)}$

generates $q\mathbb{Z}^2$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

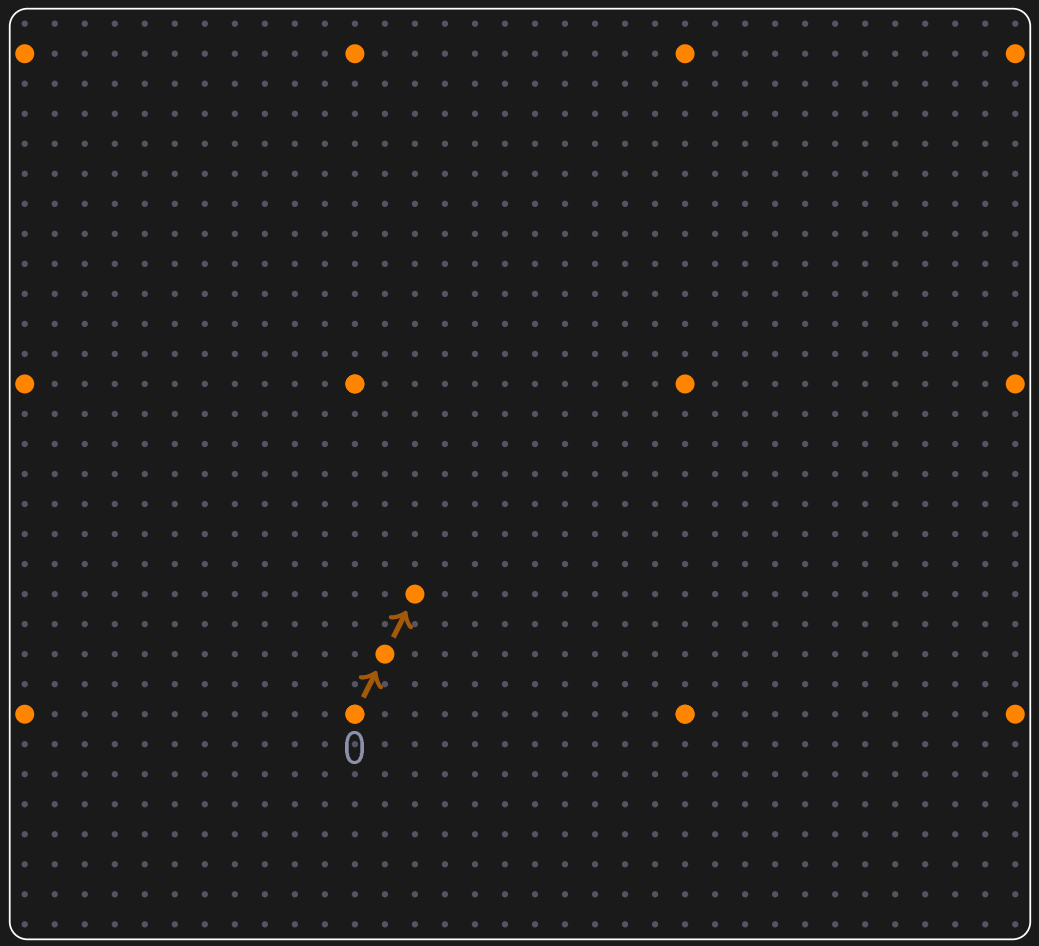
$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{generates } \mathcal{L}_h \text{ (admitted)}}$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

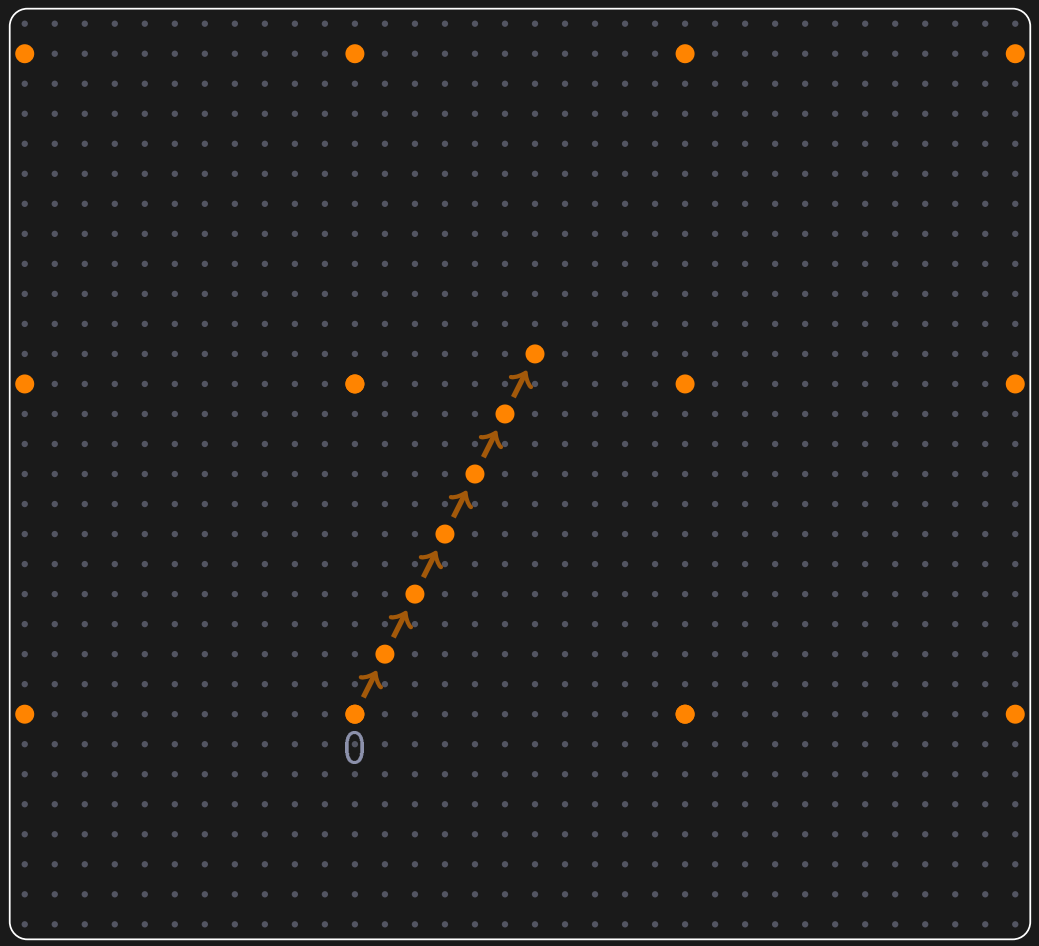
$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{generates } \mathcal{L}_h \text{ (admitted)}}$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

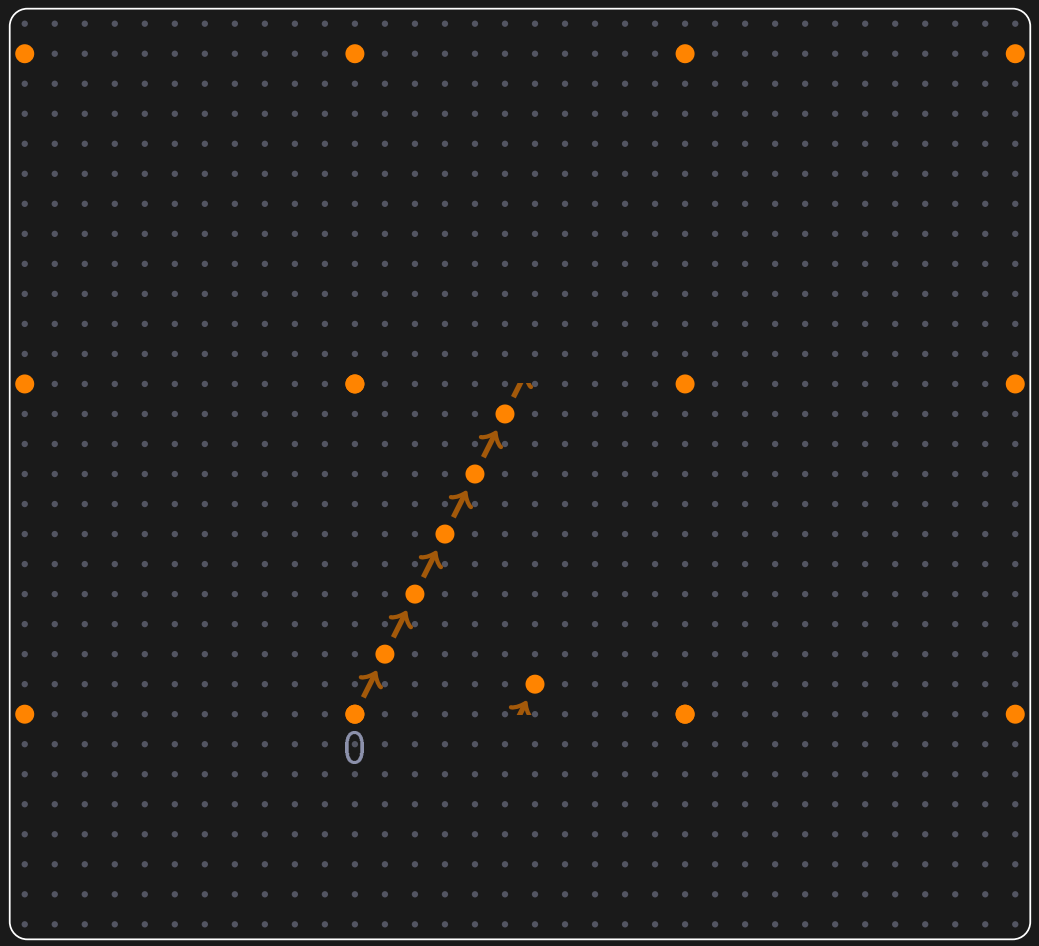
$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{generates } \mathcal{L}_h \text{ (admitted)}}$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

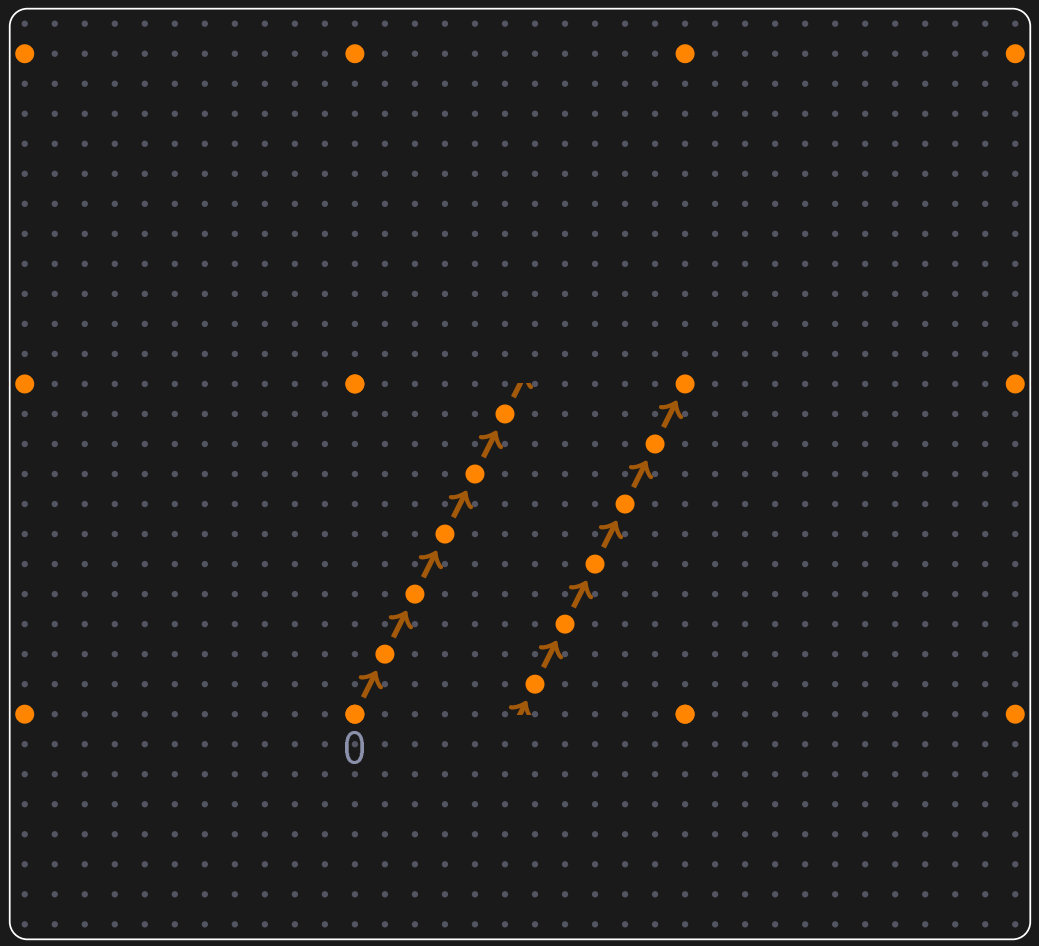
$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{generates } \mathcal{L}_h \text{ (admitted)}}$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

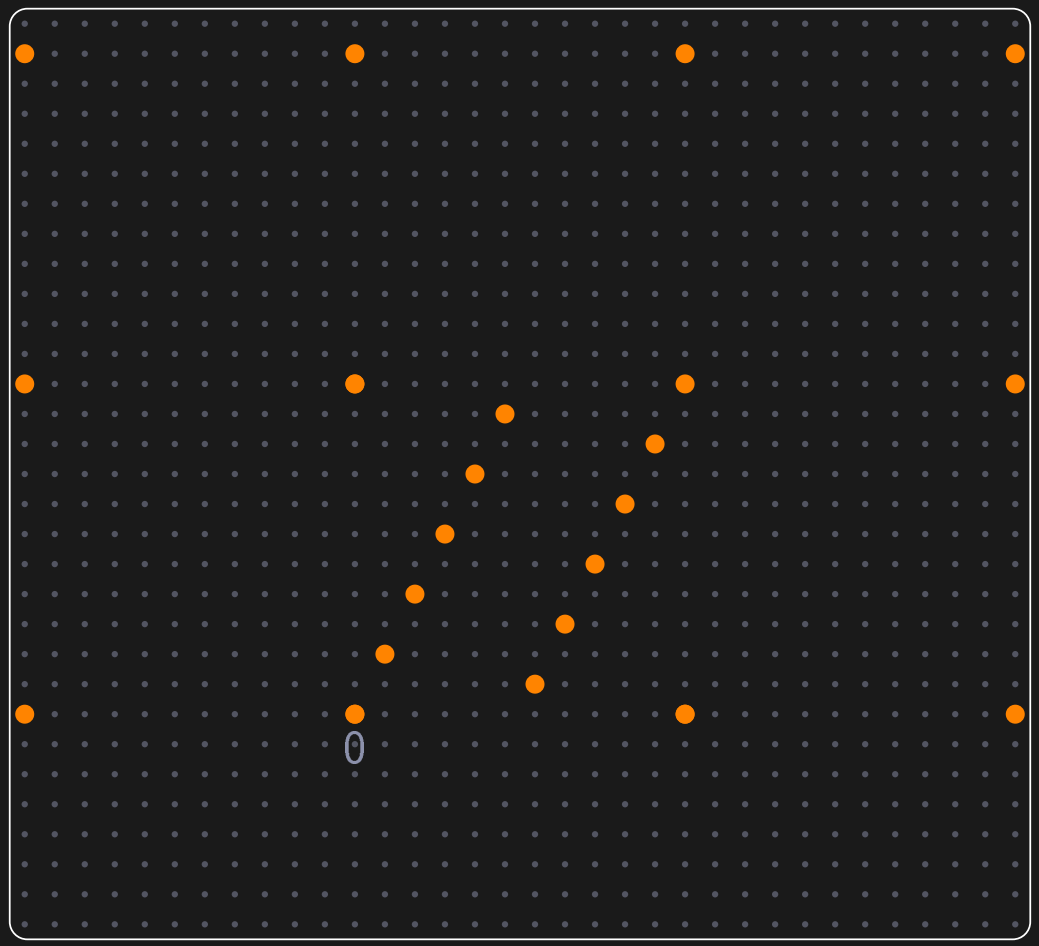
$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

generates $\mathcal{L}_h$ (admitted)

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

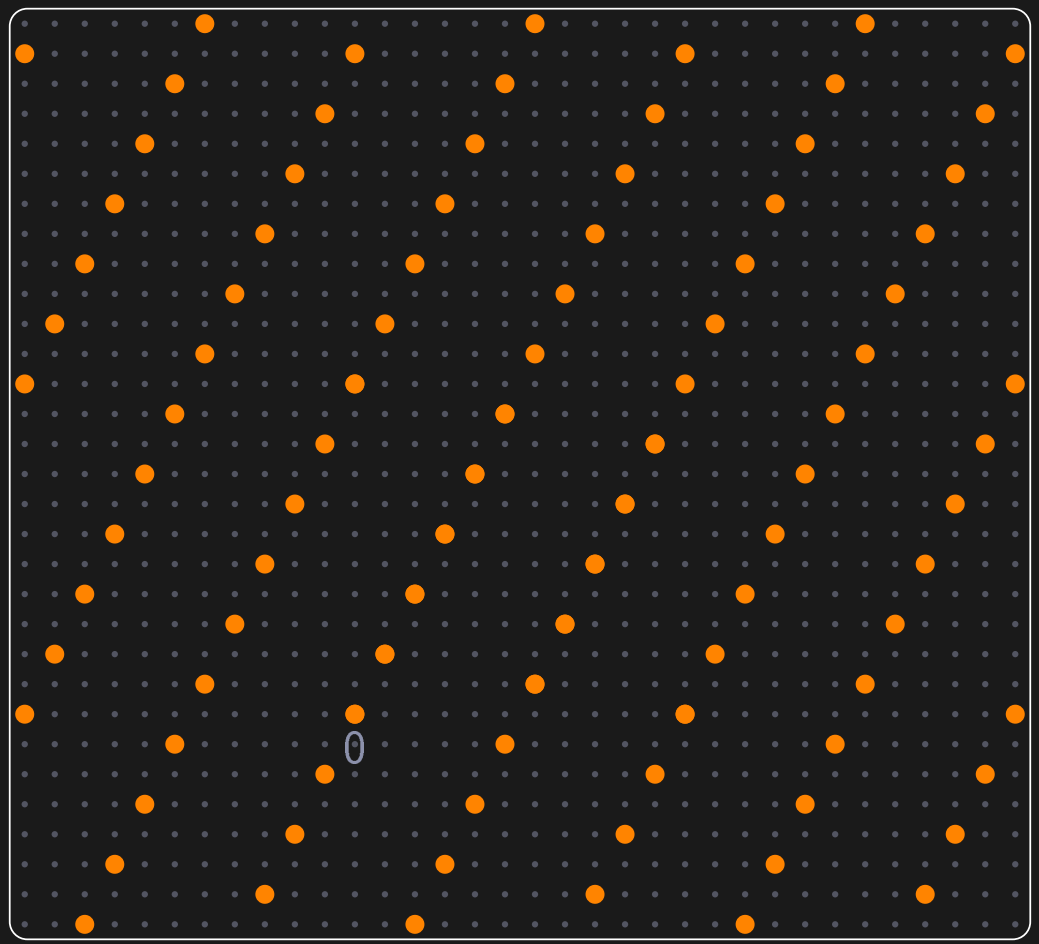
$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{generates } \mathcal{L}_h \text{ (admitted)}}$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

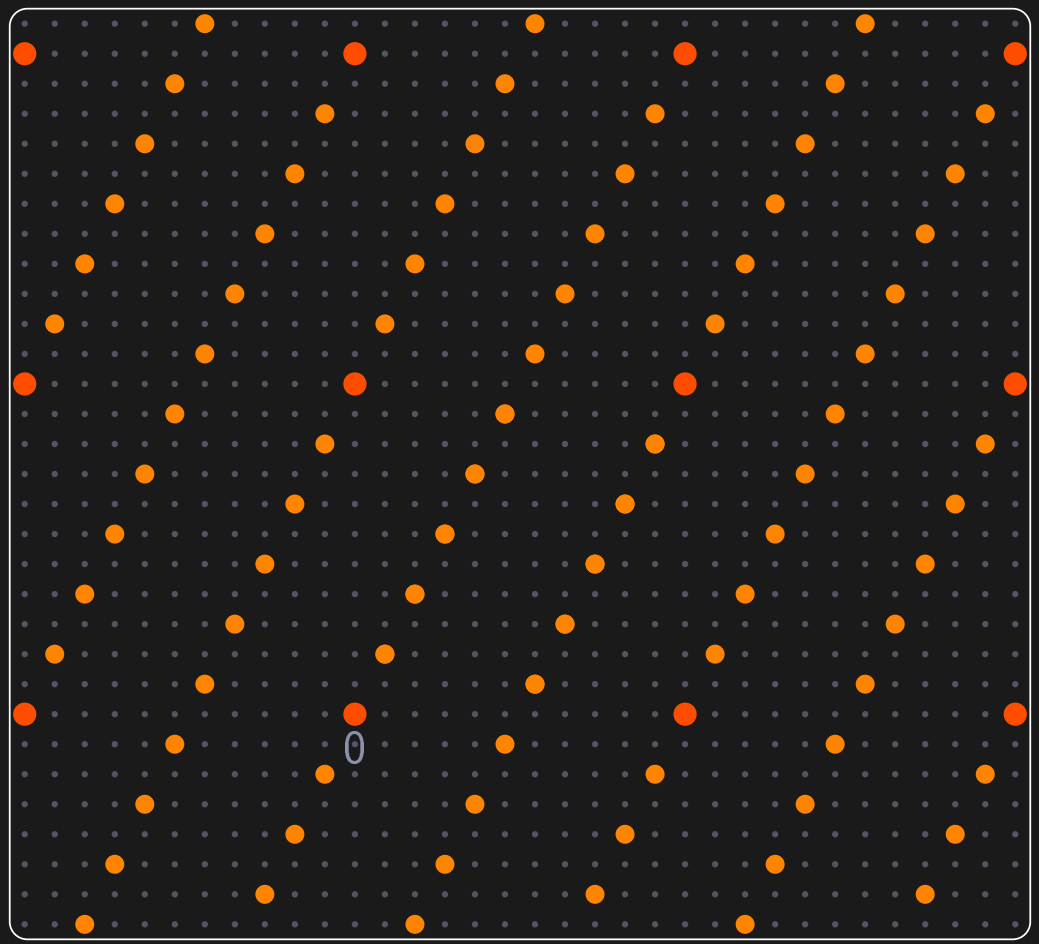
$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

generates $\mathcal{L}_h$ (admitted)

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{ (u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q \right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{generates } \mathcal{L}_h \text{ (admitted)}}$

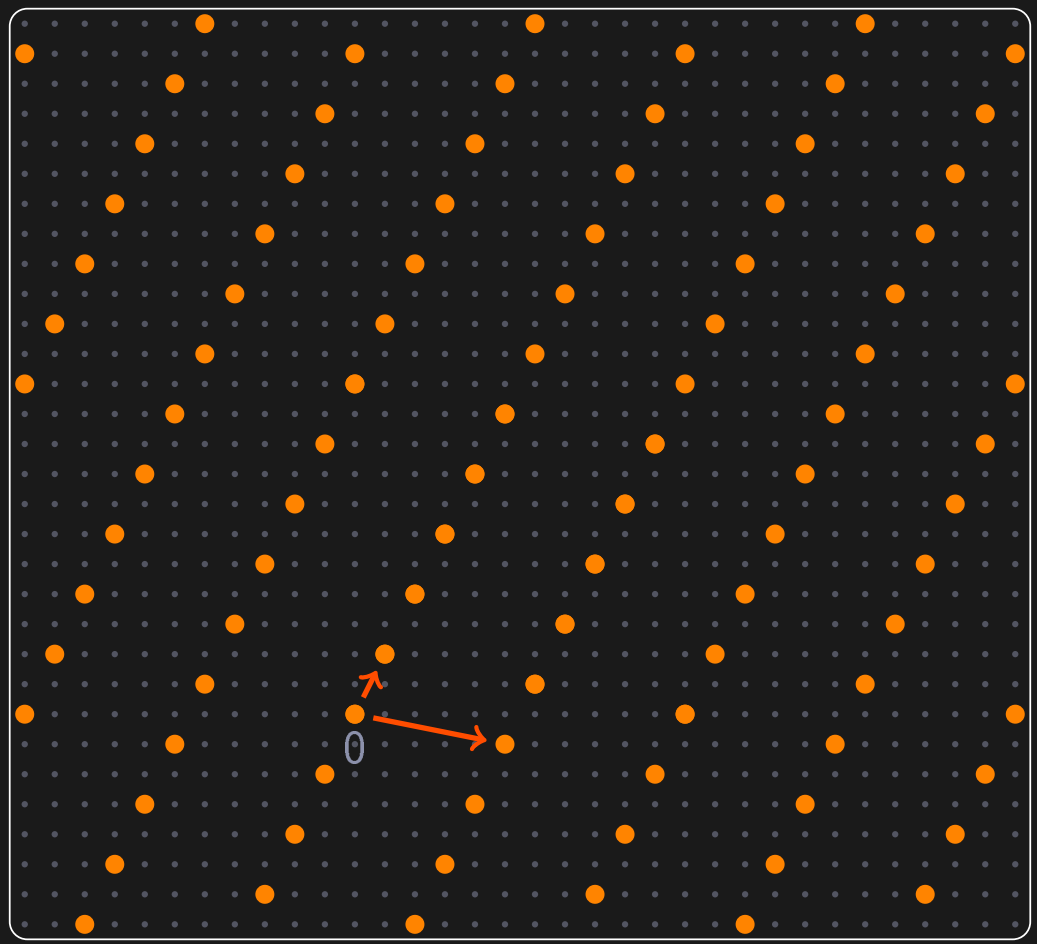
We say that

- $\mathcal{L}_h$ is q-ary $(q\mathbb{Z}^2 \subseteq \mathcal{L}_h)$

# A closer look at $\mathcal{L}_h$

Reminder: $\mathcal{L}_h = \left\{(u, v) \in \mathbb{Z}^2 \,\middle|\, vh = u \bmod q\right\}$

$\mathbb{Z}^2$ $\mathcal{L}_h$




$\mathcal{L}_h$ contains

$\underbrace{(q, 0), \quad (0, q),}_{\text{generates } q\mathbb{Z}^2} \quad (f, g)$

generates $\mathcal{L}_h$ (admitted)

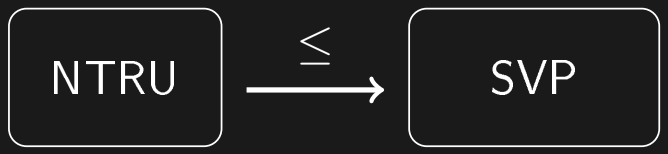
We say that

- $\mathcal{L}_h$ is q-ary $(q\mathbb{Z}^2 \subseteq \mathcal{L}_h)$
- $\mathcal{L}_h$ is unbalanced
  (its shortest basis is unbalanced)

# Back to comparisons

**Reminder:** SVP = short vector problem

We have seen

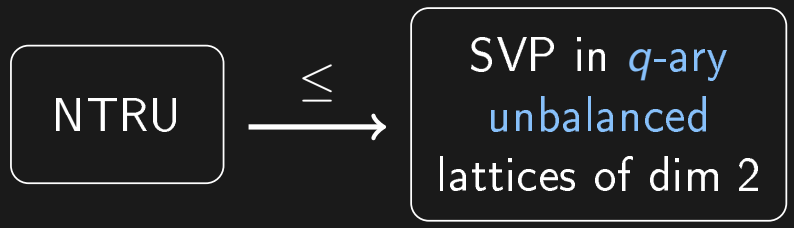



($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

# Back to comparisons

**Reminder:** SVP = short vector problem

We have seen

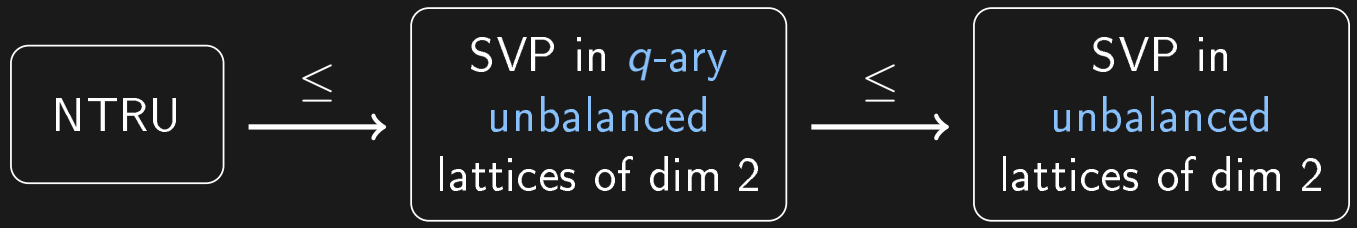



($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

# Back to comparisons

**Reminder:** SVP = short vector problem

We have seen

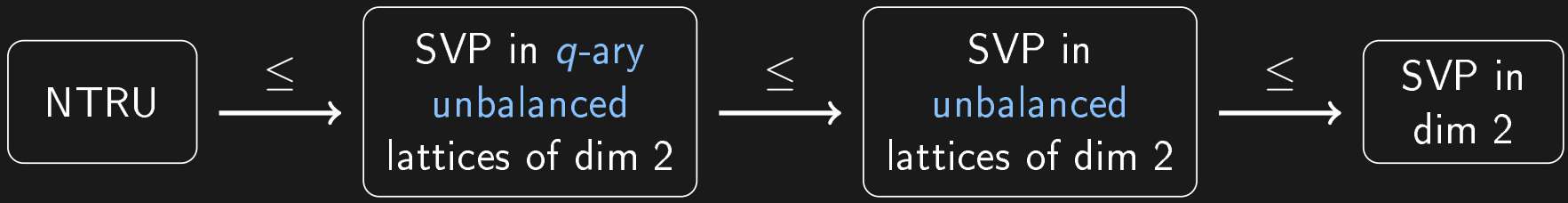



($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

# Back to comparisons

**Reminder:** SVP = short vector problem

We have seen

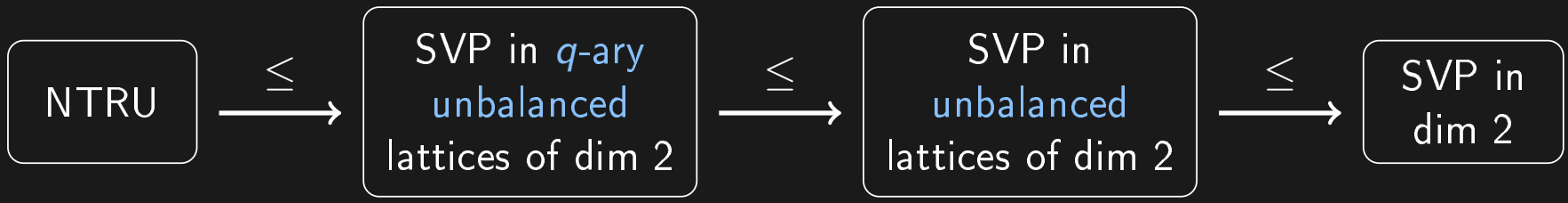



($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

# Back to comparisons

**Reminder:** SVP = short vector problem

## We have seen



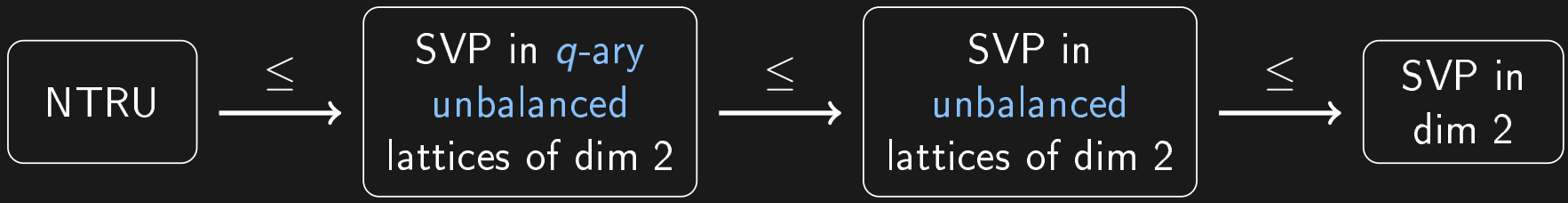

($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

## Now we can partially prove the other direction

# Back to comparisons

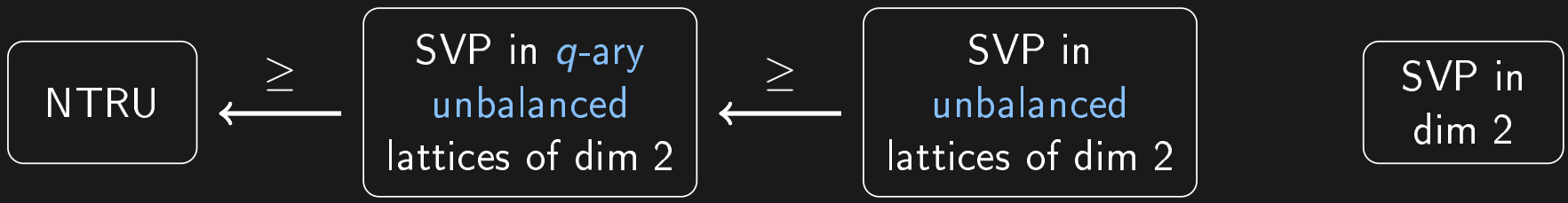
**Reminder:** SVP = short vector problem

## We have seen




($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

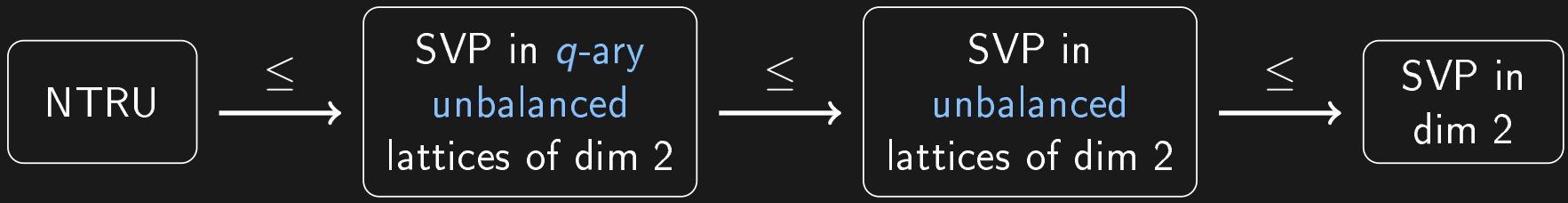
## Now we can partially prove the other direction



# Back to comparisons

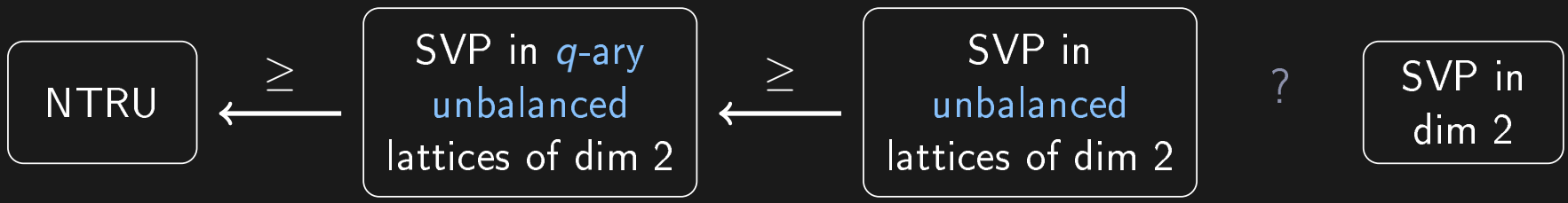
**Reminder:** SVP = short vector problem

## We have seen




($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

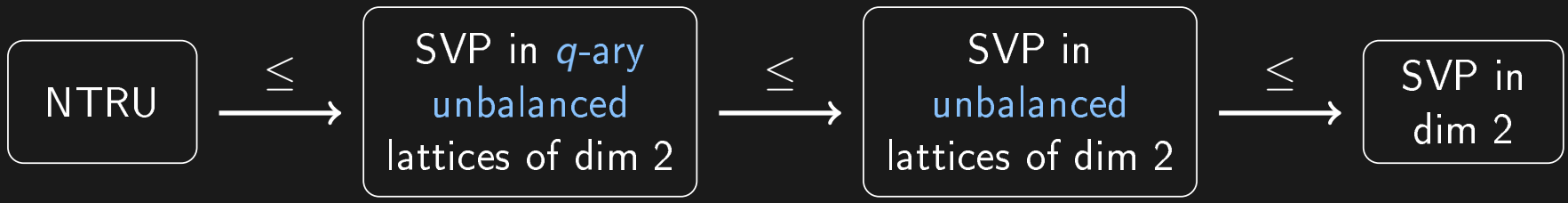
## Now we can partially prove the other direction



# Back to comparisons

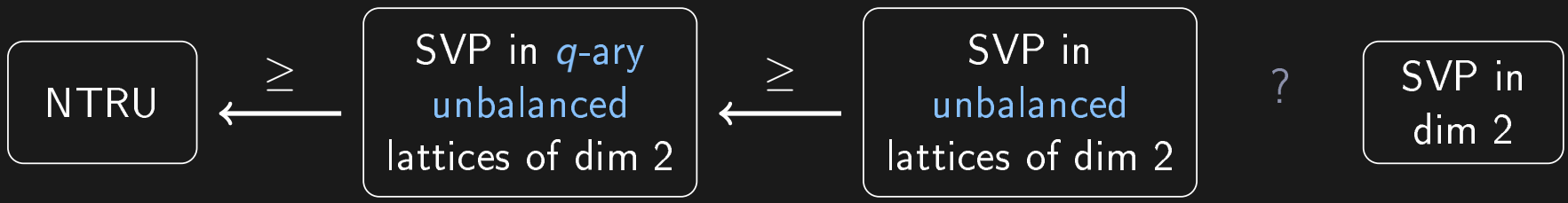
**Reminder:** SVP = short vector problem

## We have seen




($A \geq B \iff$ if you can solve $A$, you can also solve $B$)

## Now we can partially prove the other direction




NTRU $\approx$ SVP in unbalanced lattices of dim 2

# Correct definition of NTRU

**Reminder:** we said NTRU $\approx$ SVP in unbalanced lattices of dim 2

⚠ Gauss-Lagrange algorithm solves SVP in (any) lattice of dimension 2 in polynomial time ⚠

(this breaks our former definition of NTRU)

# Correct definition of NTRU

**Reminder:** we said NTRU $\approx$ SVP in unbalanced lattices of dim 2

⚠ Gauss-Lagrange algorithm solves SVP in (any) lattice of dimension 2 in polynomial time ⚠

(this breaks our former definition of NTRU)

## True NTRU definition

- $f, g \in \mathbb{Z}[X]/(X^d + 1)$ ($d = 2^\ell$, e.g., $d = 512$)
- $h = f \cdot g^{-1} \bmod q \in \mathbb{Z}_q[X]/(X^d + 1)$

# Correct definition of NTRU

**Reminder:** we said NTRU $\approx$ SVP in unbalanced lattices of dim 2

⚠ Gauss-Lagrange algorithm solves SVP in (any) lattice of dimension 2 in polynomial time ⚠

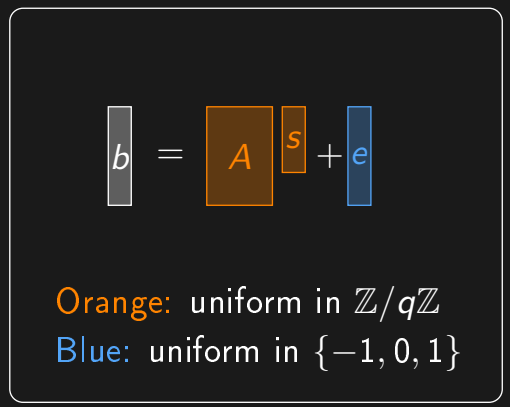
(this breaks our former definition of NTRU)

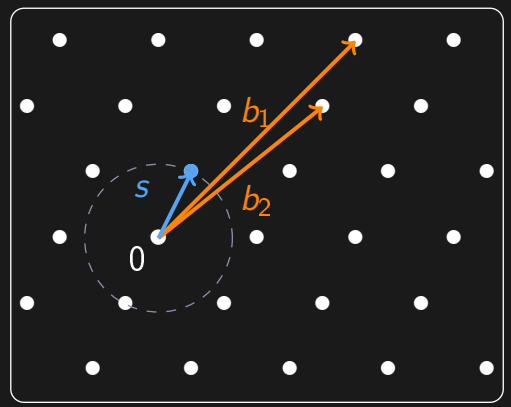
## True NTRU definition

- $f, g \in \mathbb{Z}[X]/(X^d + 1)$ ($d = 2^\ell$, e.g., $d = 512$)
- $h = f \cdot g^{-1} \bmod q \in \mathbb{Z}_q[X]/(X^d + 1)$
- $\mathcal{L}_h = \left\{(u, v) \in (\mathbb{Z}[X]/(X^d + 1))^2 \cong \mathbb{Z}^{2d} \,\middle|\, v \cdot h = u \bmod q\right\}$
  - lattice of dimension 2d (or module lattice of rank 2)

NTRU $\approx$ SVP in unbalanced module lattice of rank 2

# Conclusion


b = A s + e
Orange: uniform in $\mathbb{Z}/q\mathbb{Z}$
Blue: uniform in $\{-1, 0, 1\}$


$b_1$
$b_2$
s
0

# Conclusion

Geometric formalism is useful for

# Conclusion

Geometric formalism is useful for

- getting intuition about hardness of assumptions
  - attacks (part 2)
  - comparison with other assumptions (part 3)

# Conclusion

Geometric formalism is useful for

- getting intuition about hardness of assumptions
  - attacks (part 2)
  - comparison with other assumptions (part 3)
- getting same construction from different assumptions
  - hash-and-sign signatures (part 1)
  - Regev-like public key encryption scheme
  - other constructions? Fiat-Shamir with abort signatures? FHE? ...

# Conclusion

Geometric formalism is useful for

- getting intuition about hardness of assumptions
  - attacks (part 2)
  - comparison with other assumptions (part 3)
- getting same construction from different assumptions
  - hash-and-sign signatures (part 1)
  - Regev-like public key encryption scheme
  - other constructions? Fiat-Shamir with abort signatures? FHE? ...

Matrix formalism is useful for

- security proofs
- implementation

# Conclusion

Geometric formalism is useful for

- getting intuition about hardness of assumptions
  - attacks (part 2)
  - comparison with other assumptions (part 3)
- getting same construction from different assumptions
  - hash-and-sign signatures (part 1)
  - Regev-like public key encryption scheme
  - other constructions? Fiat-Shamir with abort signatures? FHE? ...

Matrix formalism is useful for

- security proofs
- implementation

Thank you