CS 258: Quantum Cryptography

Mark Zhandry

Previously…

Group Action

An (abelian) group action is a triple $(\mathbb{G}, \mathcal{X}, *)$ where:

- $\mathbb{G}$ is an (abelian) group
- $\mathcal{X}$ is a set
- $* : \mathbb{G} \times \mathcal{X} \to \mathcal{X}$ is an efficient binary operation satisfying

$$g * (h * x) = (gh) * x$$

- There is some element $x_0 \in \mathcal{X}$ that can be efficiently computed
- Usually ask that for each $x, y \in \mathcal{X}$, there exists a unique $g \in \mathbb{G}$ such that $y = g * x$
- Also usually ask that it is possible to efficiently identify elements of $\mathcal{X}$

Thm [Kuperberg]: Dlog in (abelian) group actions can be solved in time $2^{O(\sqrt{\log q})}$, where q is the group order

Known as “subexponential” time

Impact on cryptography

Recall: want security against attacks running in time 2^{128}

Classical groups: can in principle set group size 2^{256}

Find collision in $f(x, y) = g^x h^y$ in time $\sqrt{q}$

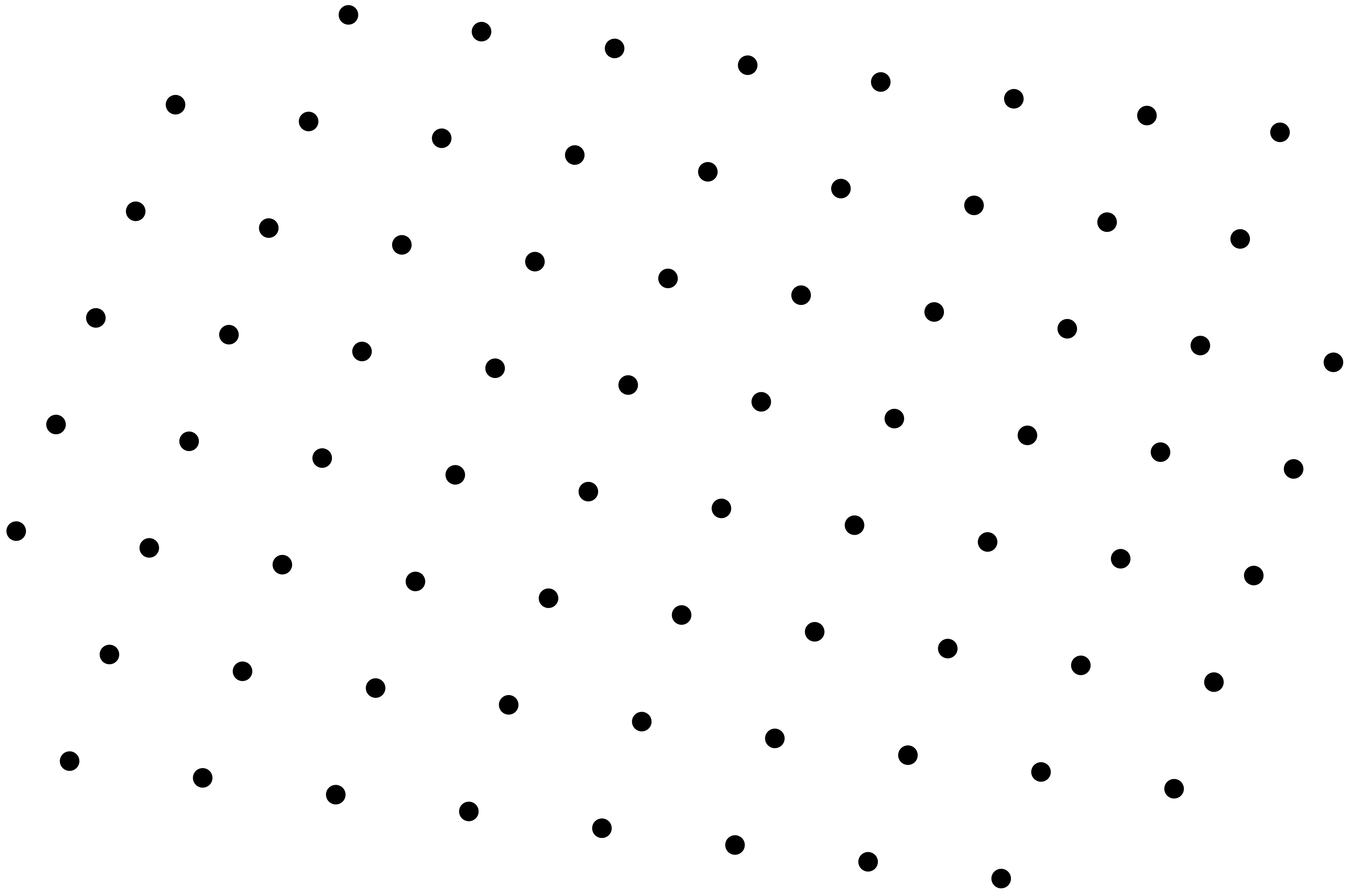
by birthday paradox

Post-quantum group actions: need groups at least $2^{128^2} \approx 2^{16384}$

Results in much less efficient schemes

Today: Lattices

Lattices



Imagine dimension in the 100s

Two equivalent descriptions of a lattice

- Discrete subgroup of $\mathbb{R}^n$

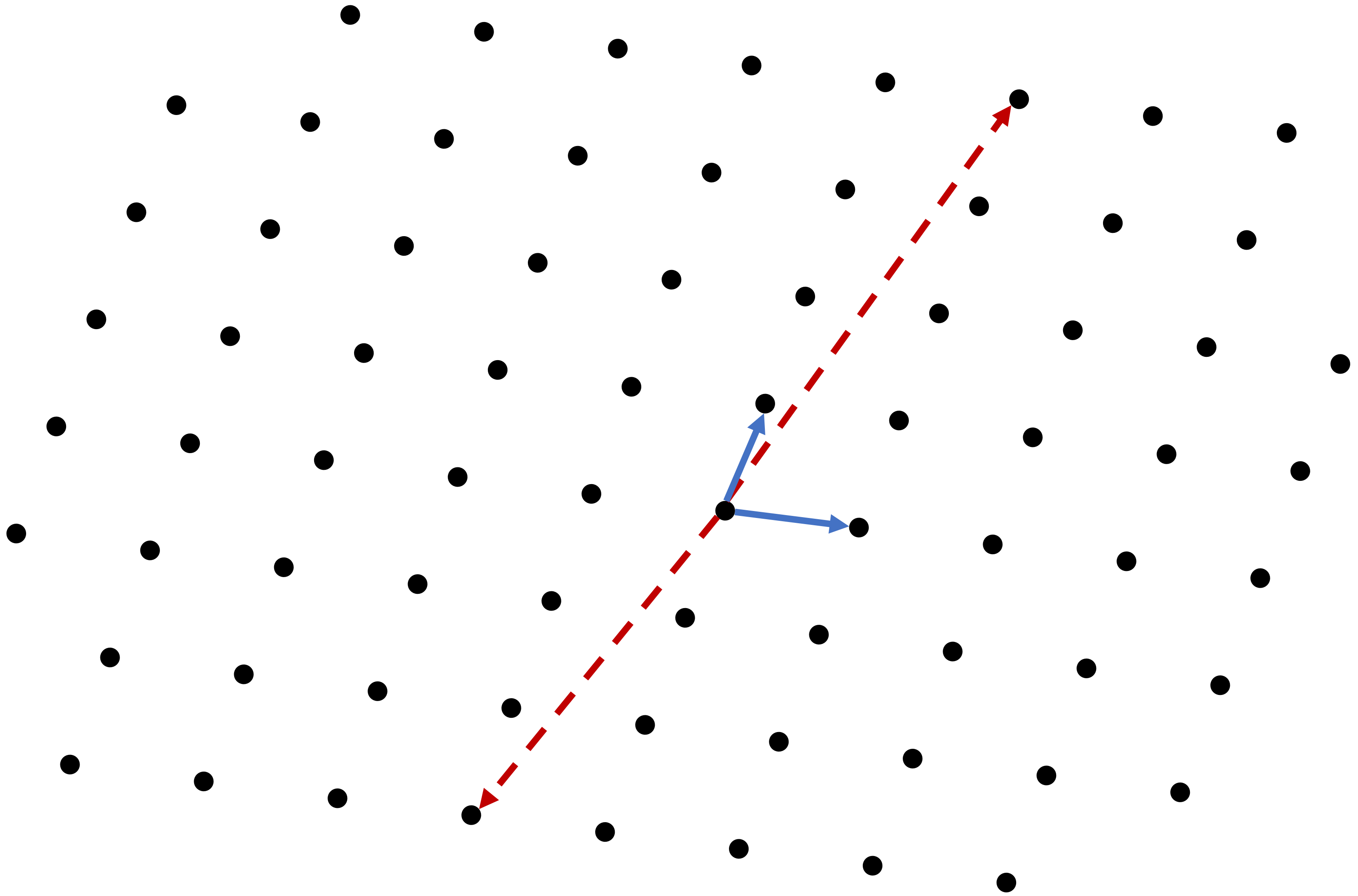
  Not a lattice: $\{a + b\sqrt{5} : a, b \in \mathbb{Z}\}$

- *Integer* linear combinations of set of vectors that are linearly independent over reals

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{B} \cdot \mathbf{v} : \mathbf{v} \in \mathbb{Z}^n\}$$

Columns of $\mathbf{B}$ are linearly independent

$\mathbf{B}$ is called a “basis” for the lattice



Different Bases

Different Bases

For vector spaces: two bases $\mathbf{B}_1, \mathbf{B}_2$ generate the same vector space if and only if there is an invertible $\mathbf{U}$ such that $\mathbf{B_2} = \mathbf{B}_1 \cdot \mathbf{U}$

For lattices: two bases $\mathbf{B}_1, \mathbf{B}_2$ generate the same lattice if and only if there is a **unimodular** $\mathbf{U}$ such that $\mathbf{B_2} = \mathbf{B}_1 \cdot \mathbf{U}$

Def: $\mathbf{U}$ is unimodular if $\mathbf{U} \in \mathbb{Z}^{n \times n}$ and $\det(\mathbf{U}) \in \{+1, -1\}$

Lemma: $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2) \Leftrightarrow \exists \mathbf{U}$ unimodular s.t. $\mathbf{B}_2 = \mathbf{B}_1 \cdot \mathbf{U}$

Proof: $\Leftarrow$, $\mathcal{L}(\mathbf{B}_2) \subseteq \mathcal{L}(\mathbf{B}_1)$

$$\mathbf{x} \in \mathcal{L}(\mathbf{B}_2) \iff \exists \mathbf{v} \in \mathbb{Z}^n : \mathbf{x} = \mathbf{B}_2 \cdot \mathbf{v}$$

$$\iff \mathbf{x} = \mathbf{B}_1 \cdot \mathbf{U} \cdot \mathbf{v} = \mathbf{B}_1 \cdot \underbrace{(\mathbf{U} \cdot \mathbf{v})}_{\in \mathbb{Z}^n}$$

$$\implies \mathbf{x} \in \mathcal{L}(\mathbf{B}_1)$$

Lemma: $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2) \Leftrightarrow \exists \mathbf{U}$ unimodular s.t. $\mathbf{B_2} = \mathbf{B_1} \cdot \mathbf{U}$

Proof: $\Leftarrow$, $\mathcal{L}(\mathbf{B}_1) \subseteq \mathcal{L}(\mathbf{B}_2)$

Claim: $\mathbf{U}$ unimodular → $\mathbf{U}^{-1}$ unimodular
Proof: Cramer's rule + $\det(\mathbf{U}) \in \{+1, -1\}$

Therefore, $\mathbf{B}_1 = \mathbf{B}_2 \cdot \mathbf{U}^{-1}$ for unimodular $\mathbf{U}^{-1}$

Proof of containment identical to before

Lemma: $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2) \Leftrightarrow \exists \mathbf{U}$ unimodular s.t. $\mathbf{B}_2 = \mathbf{B}_1 \cdot \mathbf{U}$

Proof: $\Rightarrow$

Each column of $\mathcal{L}(\mathbf{B}_2)$ contained in $\mathcal{L}(\mathbf{B}_1)$
→ $\mathbf{B}_2 = \mathbf{B}_1 \cdot \mathbf{U}$ for some $\mathbf{U} \in \mathbb{Z}^{n \times n}$

By identical argument, $\mathbf{B}_1 = \mathbf{B}_2 \cdot \mathbf{V}$ for some $\mathbf{V} \in \mathbb{Z}^{n \times n}$

Since columns are linearly independent, $\mathbf{V} = \mathbf{U}^{-1}$

$$\det(\mathbf{U}), \det(\mathbf{U}^{-1}) = \det(\mathbf{U})^{-1} \in \mathbb{Z}$$

→ $\det(\mathbf{U}) \in \{+1, -1\}$

Determinant of lattice

For full-rank lattices, $\det(\mathcal{L}) = |\det(\mathbf{B})|$, for any basis $\mathbf{B}$

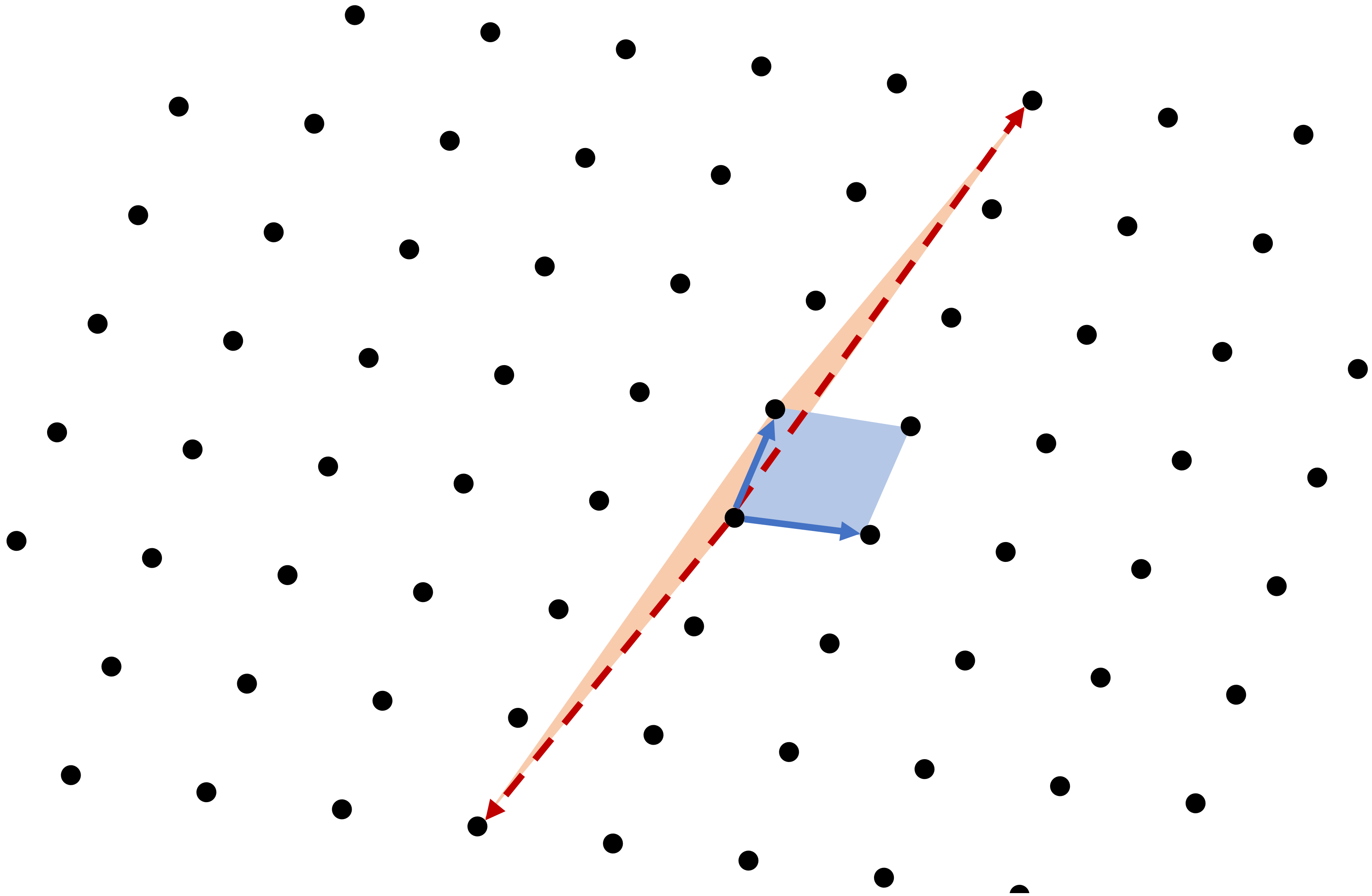
Lemma: determinant independent of basis

Proof: if $\mathbf{B}_2 = \mathbf{B}_1 \cdot \mathbf{U}$ for unimodular $\mathbf{U}$

$$\det(\mathbf{B}_1) = \det(\mathbf{B}_2)\det(\mathbf{U}) = \det(\mathbf{B}_2)$$

For general lattices, $\det(\mathcal{L}) = \sqrt{\det(\mathbf{B}^T\mathbf{B})}$

Determinant of lattice



Measure of how dense the lattice is

Full-rank lattice: $\mathsf{span}(\mathbf{B}) = \mathbb{R}^n \iff \mathbf{B} \in \mathbb{R}^{n \times n}$

Integer lattice: $\mathbf{B} \in \mathbb{Z}^{m \times n}$

We will generally consider only full-rank integer lattices

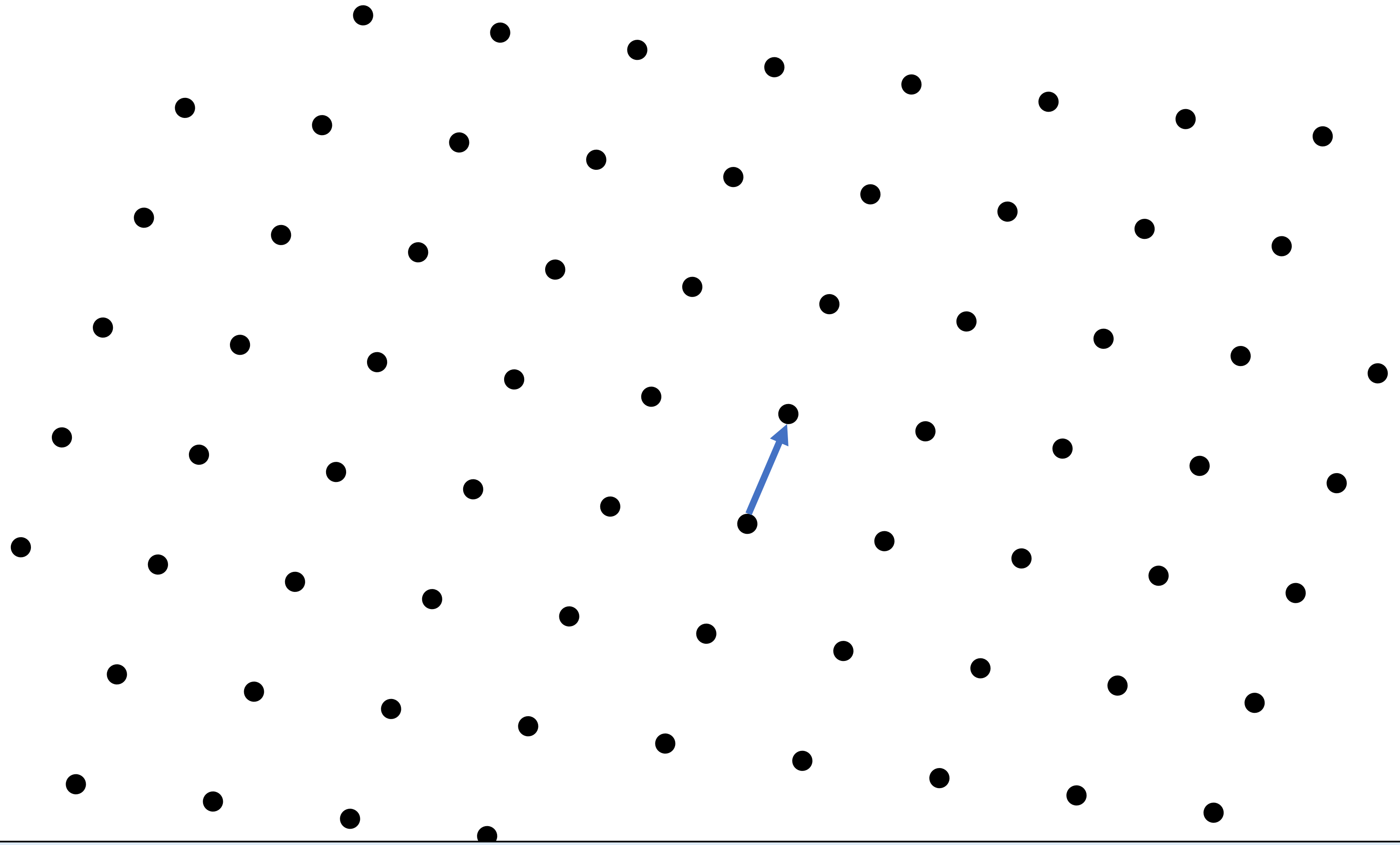
Note that for integer lattices, can consider spanning set that is not full-rank, and still guarantee discreteness

Hard problems on lattices

Shortest vector problem (SVP)

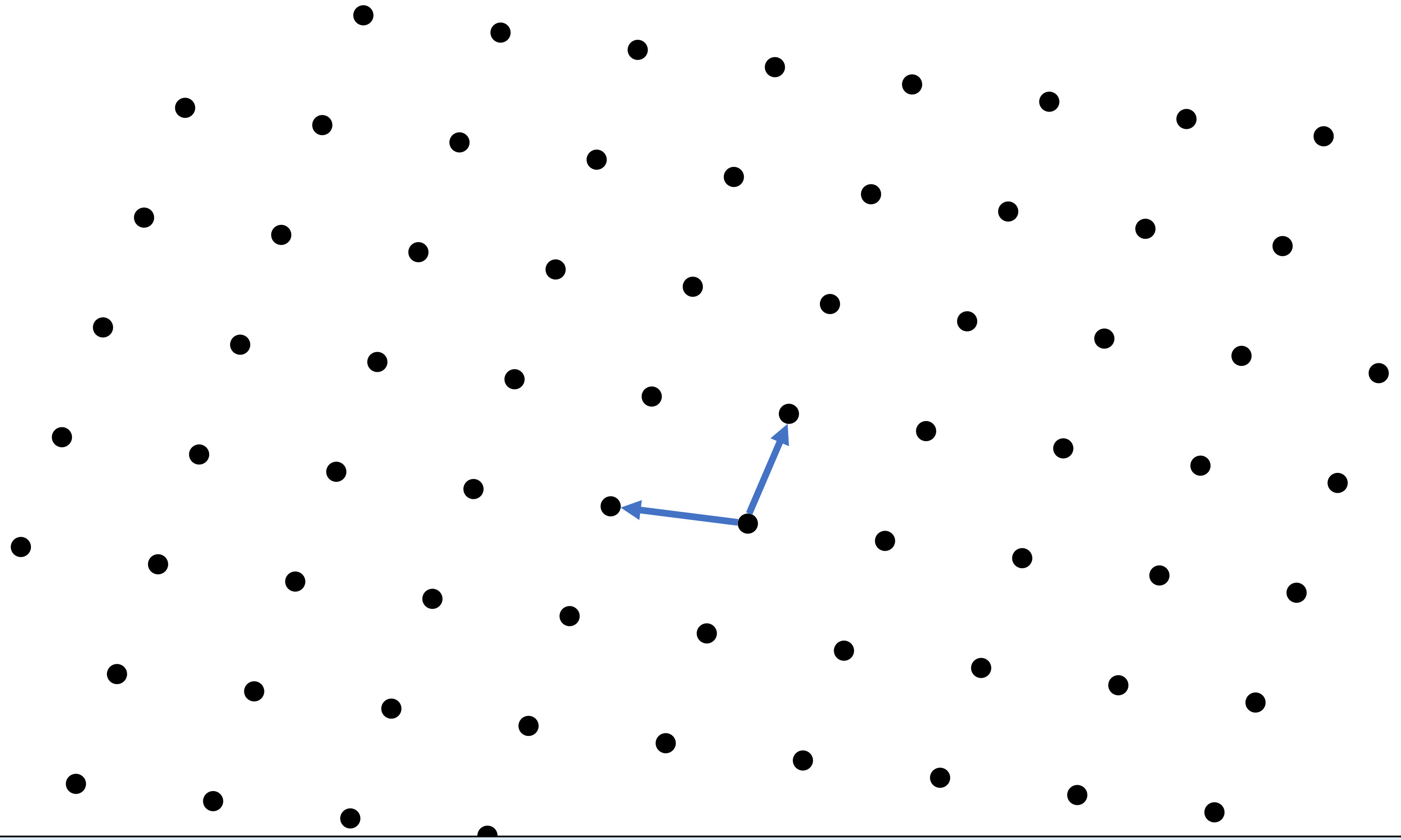
Closes vector problem (CVP)

SVP



(Approx.) shortest vector problem (SVP): given lattice (described by some basis), find (approx.) shortest vector

SIVP



(Approx.) shortest independent vector problem (SVP): given lattice (described by some basis), find (approx.) shortest basis

S(I)VP in dimension 1 is easy

A basis for a dimension-1 lattice is just a scalar $\mathbf{B} = b \in \mathbb{R}$

Only possible bases are $\pm b$

Bases are already shortest "vector"

S(I)VP in dimension 1 is easy

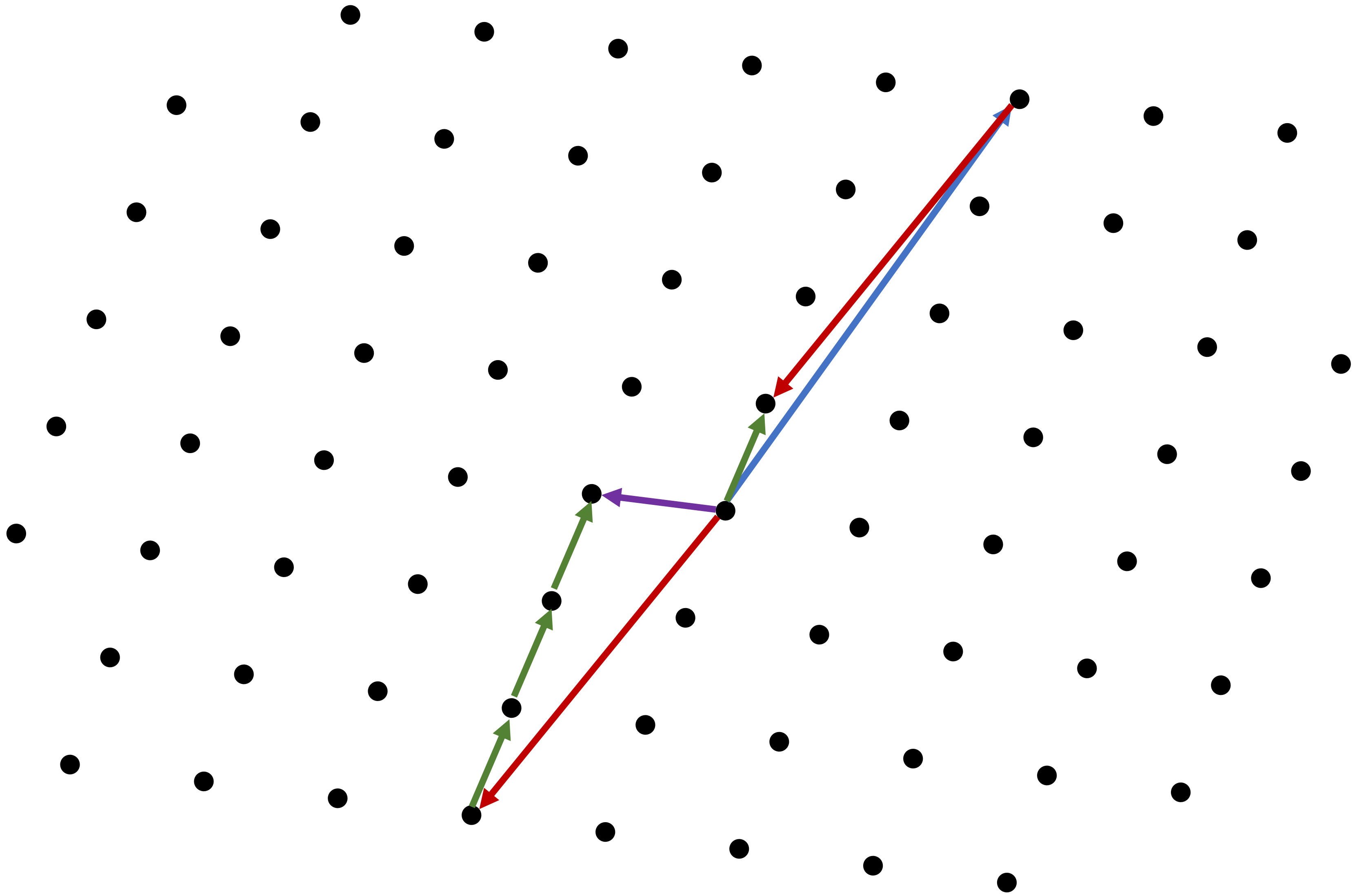
A slightly less trivial example:

Let $a, b \in \mathbb{Z}$, find basis for lattice generated by a, b

Solution: $\mathbf{B} = \mathsf{GCD}(a, b)$

Algorithm: subtract from larger element multiples of smaller element until larger element is smaller. Terminate when smaller element is 0

S(I)VP in dimension 2 is easy



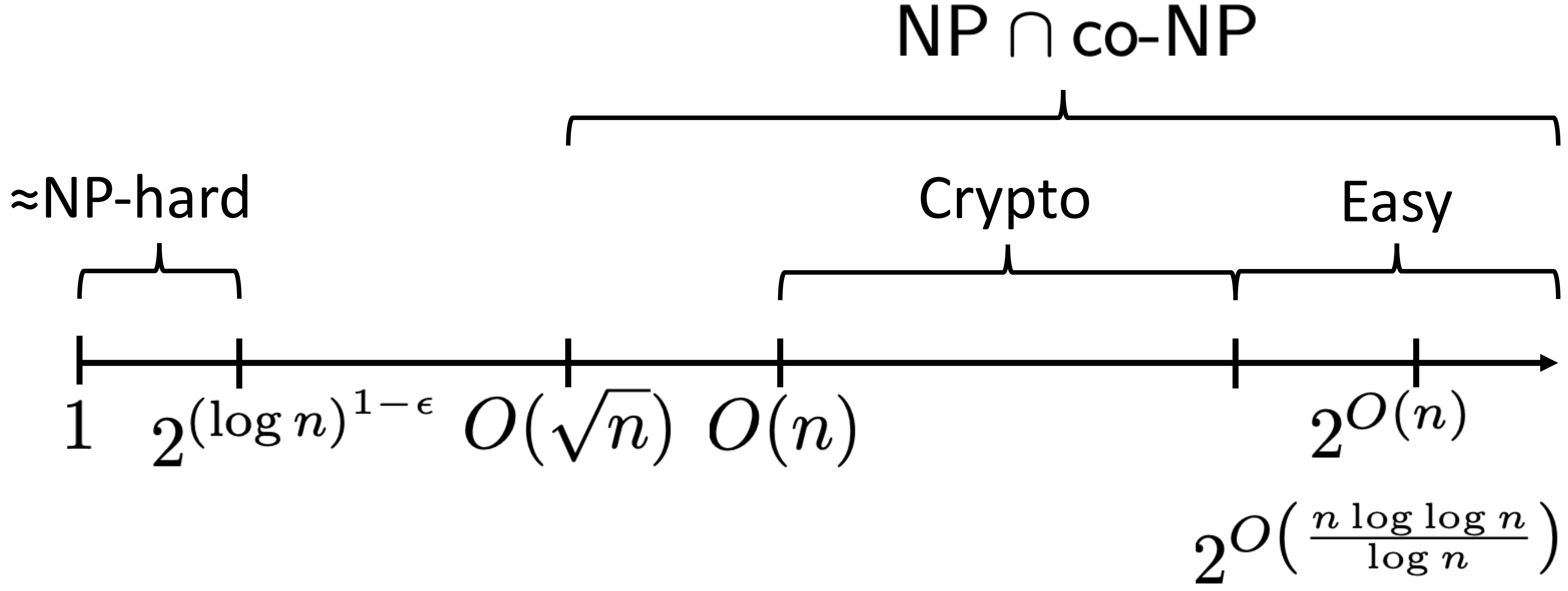
2-dimensional version of GCD

Generalization of GCD to higher-dimensions is called LLL (Lenstra–Lenstra–Lovász)

In higher dimensions, especially beyond dimension 5, LLL fails to give shortest vector

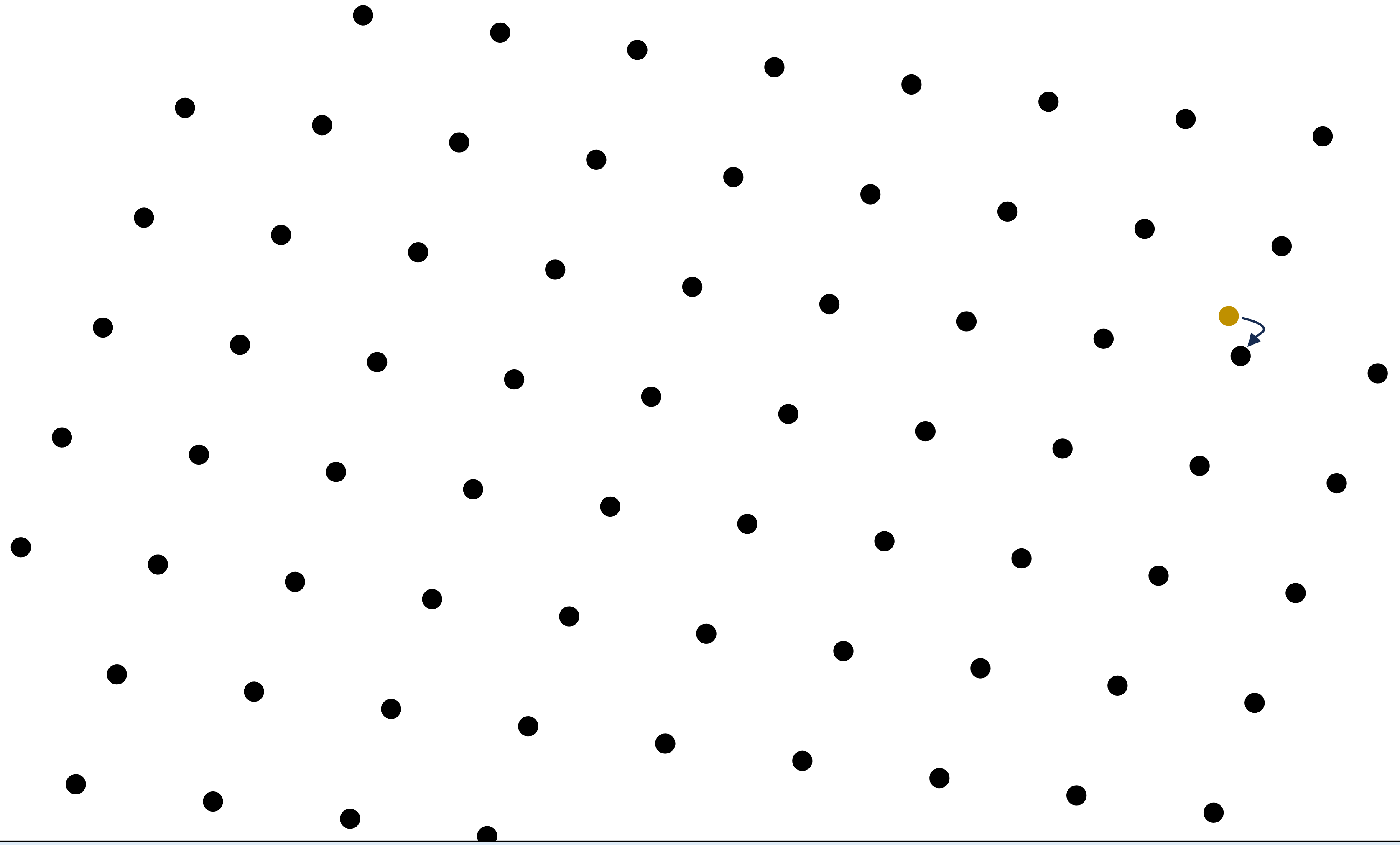
It does give a ”reasonably short” basis (within factor $2^{O(n)}$ of optimal)

Hardness of SVP



Approximation ratio

CVP



(Approx.) closest vector problem (CVP): given lattice and point off lattice, find (approx.) closest lattice point

We’ve actually seen lattices before

Let $f : \mathbb{Z}^n \to \mathcal{X}$ be a periodic function

The set of periods is a lattice!

Given Shor’s algorithm, no hope of hiding the description of the period as a lattice

SVP: finding a short period. Seems hard even for quantum

Historically, lattices (specifically LLL) were used for cryptanalysis (breaking crypto)

However, in 1990's hard problems on lattices emerged as a potential tool for cryptography, can solve many problems we don't otherwise no how to solve

With looming threat of quantum computers, now arguably main focus for post-quantum cryptosystems

An easy lattice: $\mathbb{Z}^n$

SIVP: the standard basis vectors

CVP: round each coordinate

Measure of good bases

Intuition: SVP and CVP are easy in $\mathbb{Z}^n$ because we have a really good basis, namely the standard basis

For a general lattice, (approximate) SVP and CVP will be easy if we have a basis under which $\mathcal{L}$ ”looks like” $\mathbb{Z}^n$

Roughly, want basis vectors to be approximately orthogonal

Since determinant is preserved, this correlates with basis vectors being “short”

Gram-Schmidt Orthogonalization

(no normalization)

$$\mathbf{B} = \left(\; \mathbf{b}_1 \; \middle| \; \mathbf{b}_2 \; \middle| \; \cdots \; \right)$$

$$\tilde{\mathbf{b}}_1 = \mathbf{b}_1$$

$$\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - \frac{\tilde{\mathbf{b}}_1 \cdot \mathbf{b}_2}{|\tilde{\mathbf{b}}_1|^2} \tilde{\mathbf{b}}_1$$

Note: $\tilde{\mathbf{b}}_i$ not in lattice

$$\tilde{\mathbf{b}}_3 = \mathbf{b}_3 - \frac{\tilde{\mathbf{b}}_1 \cdot \mathbf{b}_3}{|\tilde{\mathbf{b}}_1|^2} \tilde{\mathbf{b}}_1 - \frac{\tilde{\mathbf{b}}_2 \cdot \mathbf{b}_3}{|\tilde{\mathbf{b}}_2|^2} \tilde{\mathbf{b}}_2$$

...

Gram-Schmidt Orthogonalization
(no normalization)

$$\mathbf{B} = \left(\ \mathbf{b}_1 \ \middle| \ \mathbf{b}_2 \ \middle| \ \cdots \ \right)$$

$$\tilde{\mathbf{B}} = \left(\ \tilde{\mathbf{b}}_1 \ \middle| \ \tilde{\mathbf{b}}_2 \ \middle| \ \cdots \ \right)$$

$$\det(\mathbf{B}) = \det(\tilde{\mathbf{B}})$$

A good basis is therefore one where $\tilde{\mathbf{B}} \approx \mathbf{B}$

CVP with a good basis

Babai's nearest plane

Given basis $\mathbf{B}$ and a target $\mathbf{c}$, do the following:

$$\mathbf{c}' \leftarrow \mathbf{c}$$

For $i = n, \cdots, 1, \mathbf{c}' \leftarrow \mathbf{c}' - \left\lceil \frac{\tilde{\mathbf{b}}_i \cdot \mathbf{c}'}{|\tilde{\mathbf{b}}_i|^2} \right\rfloor \mathbf{b}_i$

Output $\mathbf{c} - \mathbf{c}'$

Intuition: each update to $\mathbf{c}'$ is trying to get it as close to the origin as possible while only adding/subtracting lattice points

$\mathbf{c} - \mathbf{c}'$ always stays a lattice vector, and $\mathbf{c}'$ small

➡ Decent CVP solution

Lemma: $|\mathbf{c}'|^2 \leq \frac{1}{4}\sum_i |\tilde{\mathbf{b}}_i|^2$

Proof: rotate lattice so that $\tilde{\mathbf{b}}_i/|\tilde{\mathbf{b}}_i|$ are standard basis vectors

After first update $\mathbf{c}' \leftarrow \mathbf{c}' - \left\lceil \frac{\tilde{\mathbf{b}}_n \cdot \mathbf{c}'}{|\tilde{\mathbf{b}}_n|^2} \right\rfloor \mathbf{b}_n$,

last coordinate is range $\left[-|\tilde{\mathbf{b}}_n|/2, |\tilde{\mathbf{b}}_n|/2\right]$

Future updates do not change last coordinate

Lemma: $|\mathbf{c}'|^2 \leq \frac{1}{4}\sum_i |\tilde{\mathbf{b}}_i|^2$

Proof: Applying argument to each coordinate shows that coordinate i ends up in range

$$\left[-|\tilde{\mathbf{b}}_i|/2, |\tilde{\mathbf{b}}_i|/2\right]$$

Norm of ith coordinate of final $\mathbf{c}'$ bounded by $|\tilde{\mathbf{b}}_i|/2$

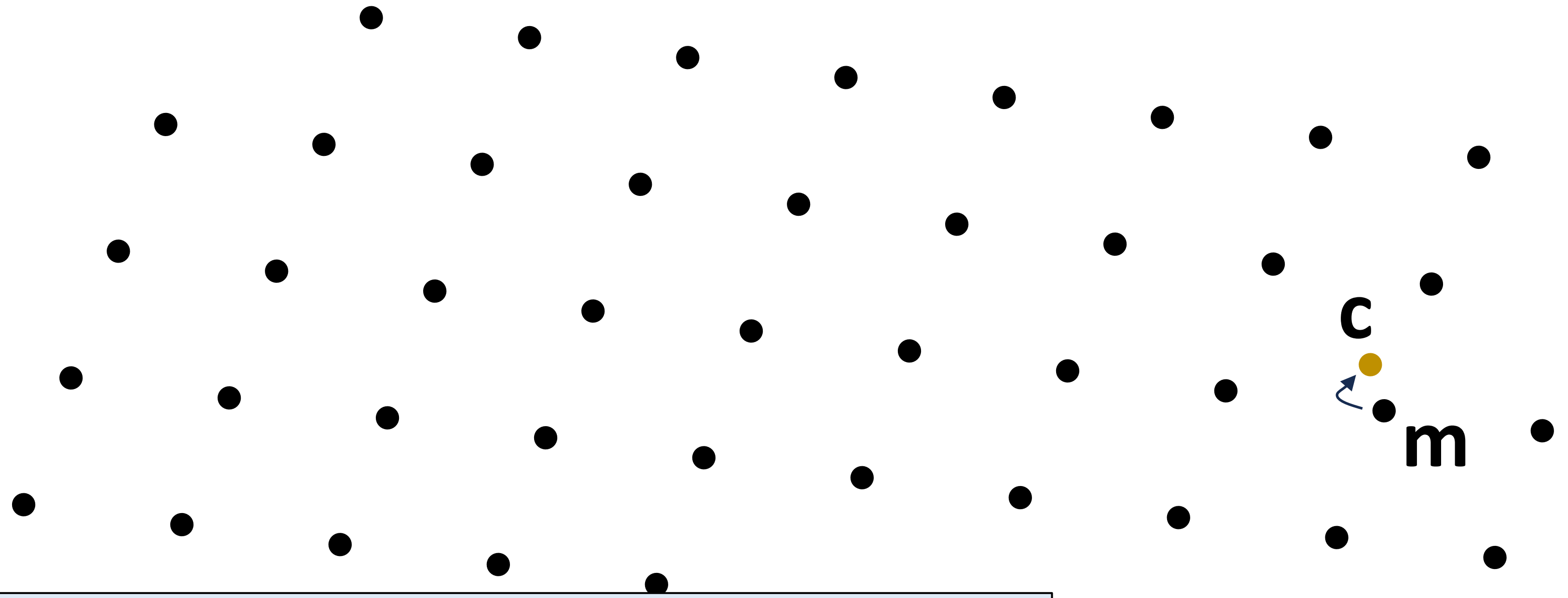
→ Norm bound follows from Pythagorean theorem

Notion of good bases and bad bases great for cryptography:

Good basis = secret key

Bad basis = public key

Encryption from lattices



Encrypt m:

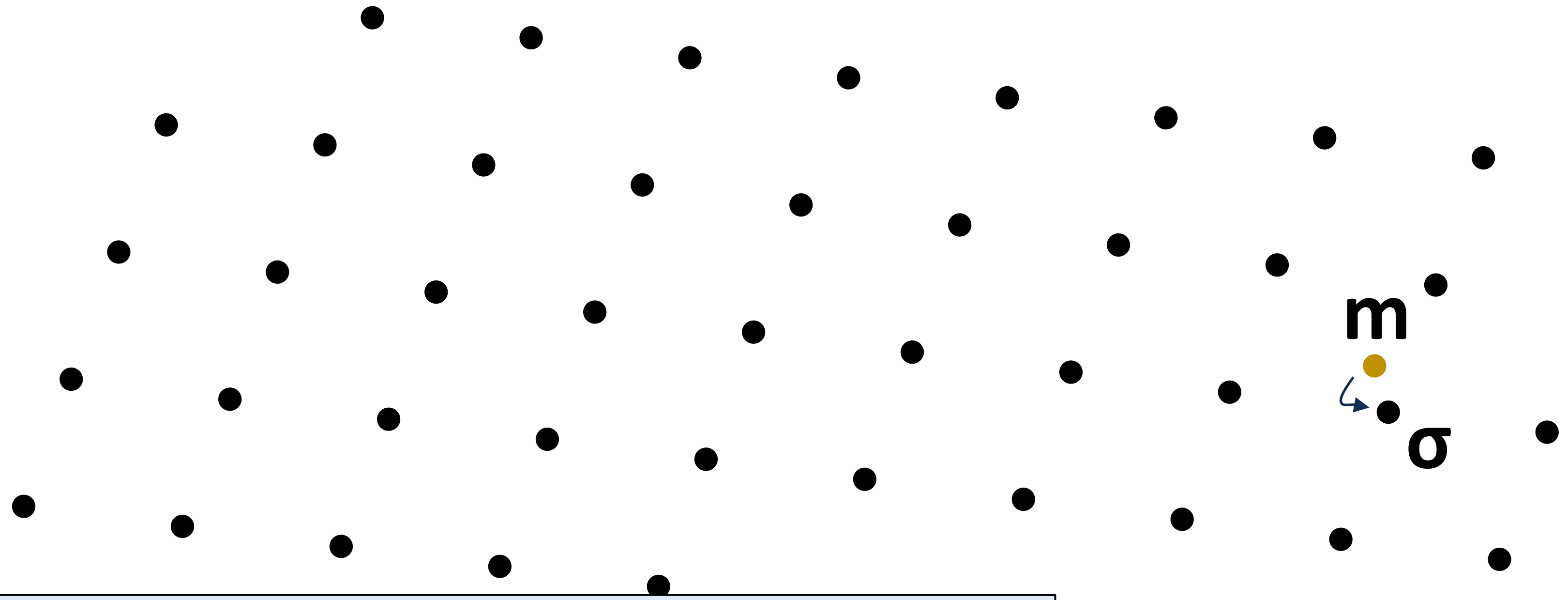
(1) Map **m** to lattice point

(2) Output close non-lattice point

Decrypt c: use good basis + Babai

Security intuitively relies on hardness of CVP given bad basis

Signatures from lattices



Sign **m**:

(1) Map **m** to non-lattice point

(2) Output close lattice point

Verifiy **m, σ**: Check closeness and that **σ** in lattice

Security intuitively relies on hardness of CVP given bad basis

Next time:

SIS and LWE: (approx.) SVP and CVP for a special family of lattices