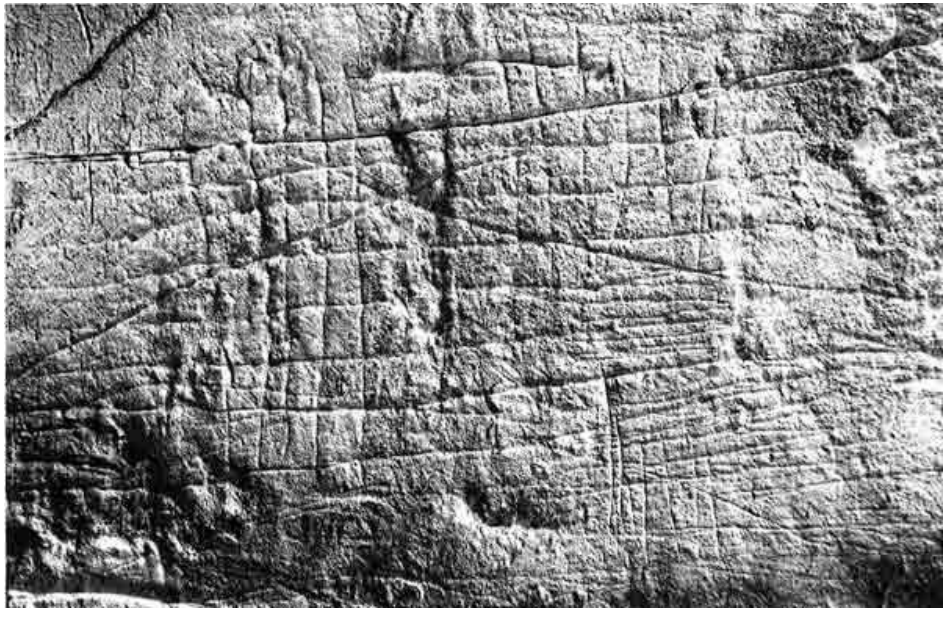
The prehistory of lattice-based cryptanalysis

Antoine Joux

Fondation UPMC, Sorbonne Universités/UPMC/LIP6/Almasty

July 6th, 2016, Workshop HEAT

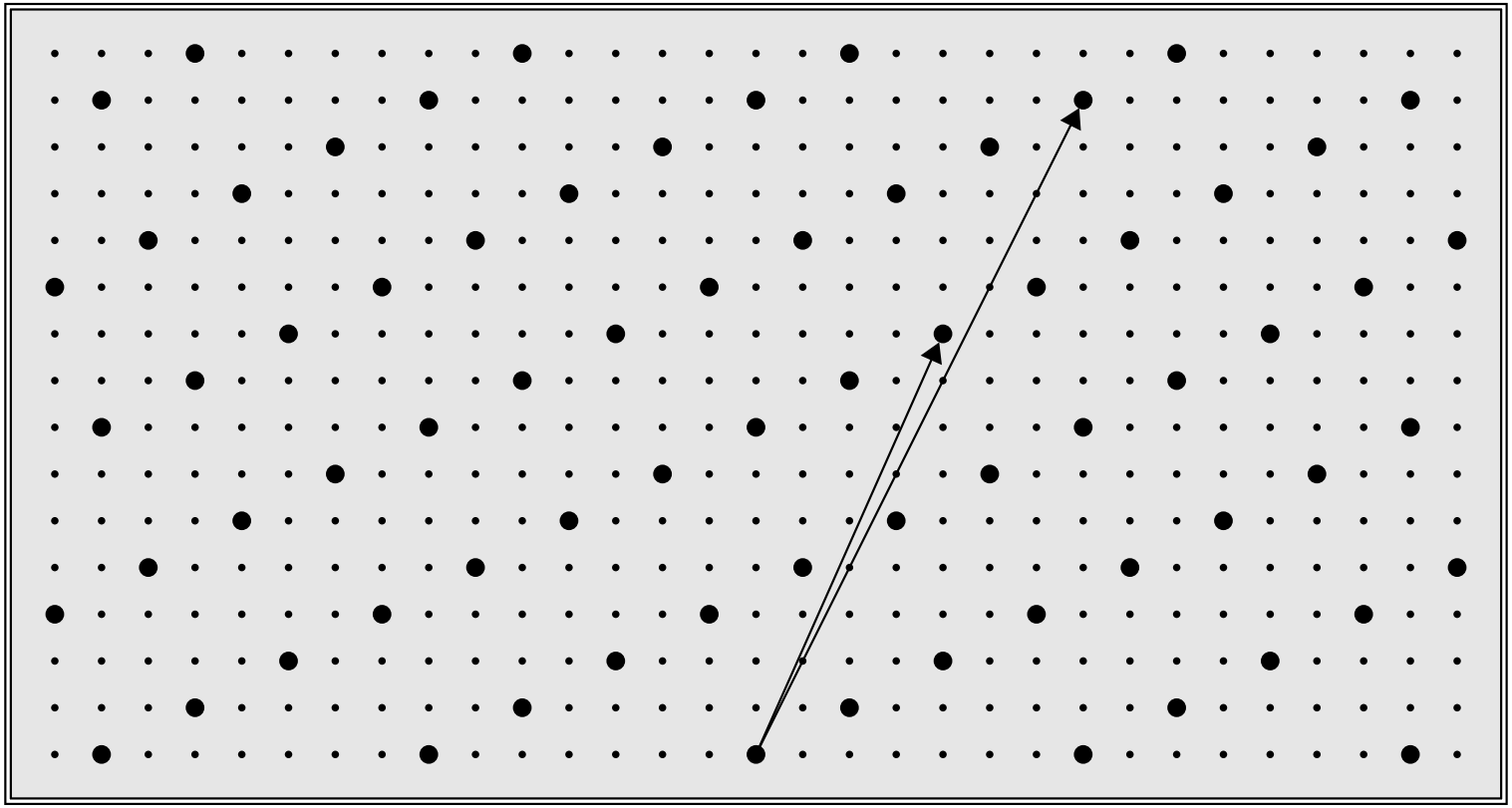


Basics of lattice reduction

Lattices

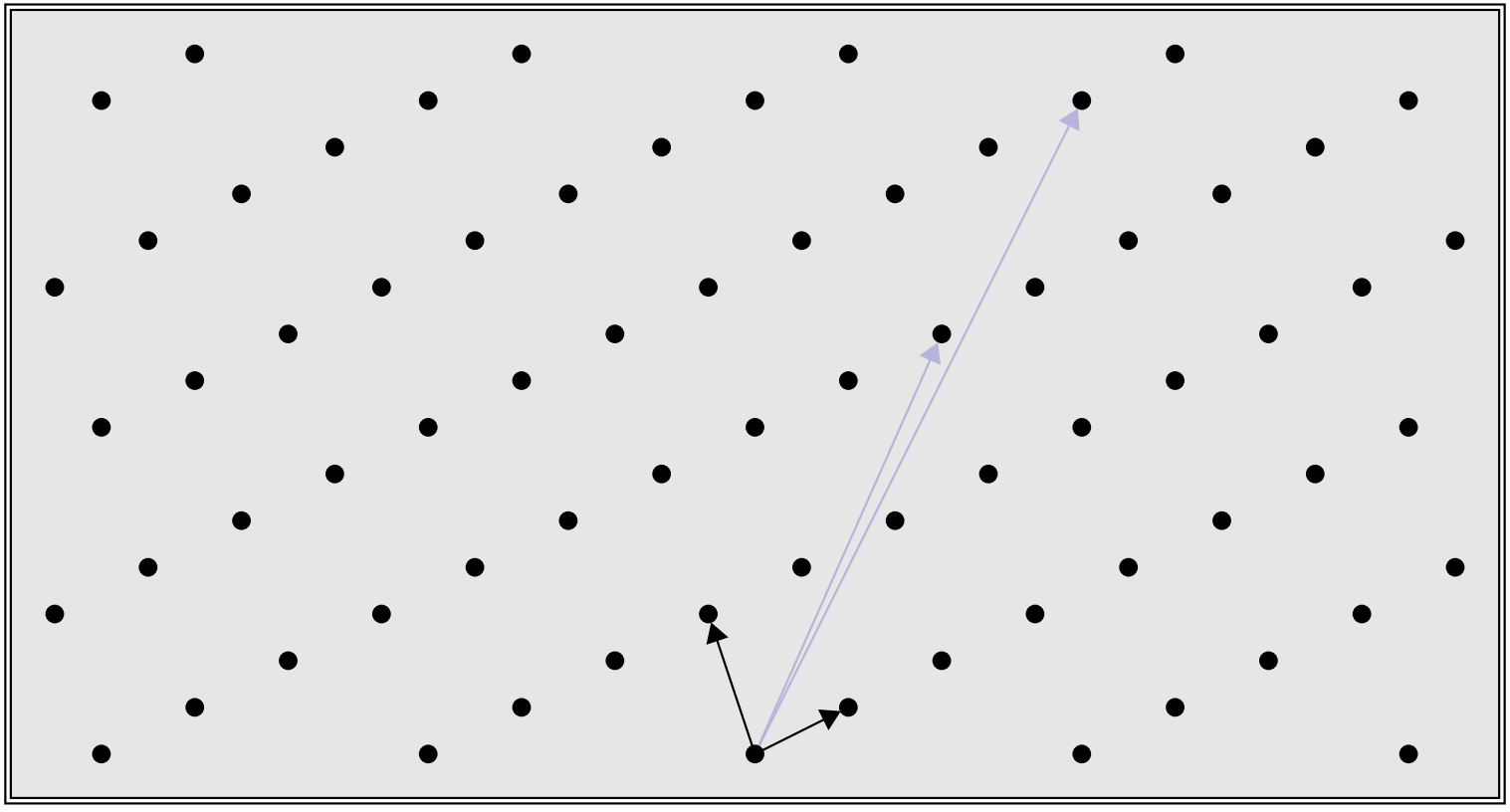
- A lattice is a discrete subgroup of $\mathbb{R}^n$
- Equivalently, set of integral linear combinations:

$$\alpha_1 \vec{b_1} + \cdots + \alpha_n \vec{b_m} \quad \text{with } m \leq n$$



Lattice reduction

- Lattice reduction looks for a “good” basis
- Easy to visualize in dimension 2

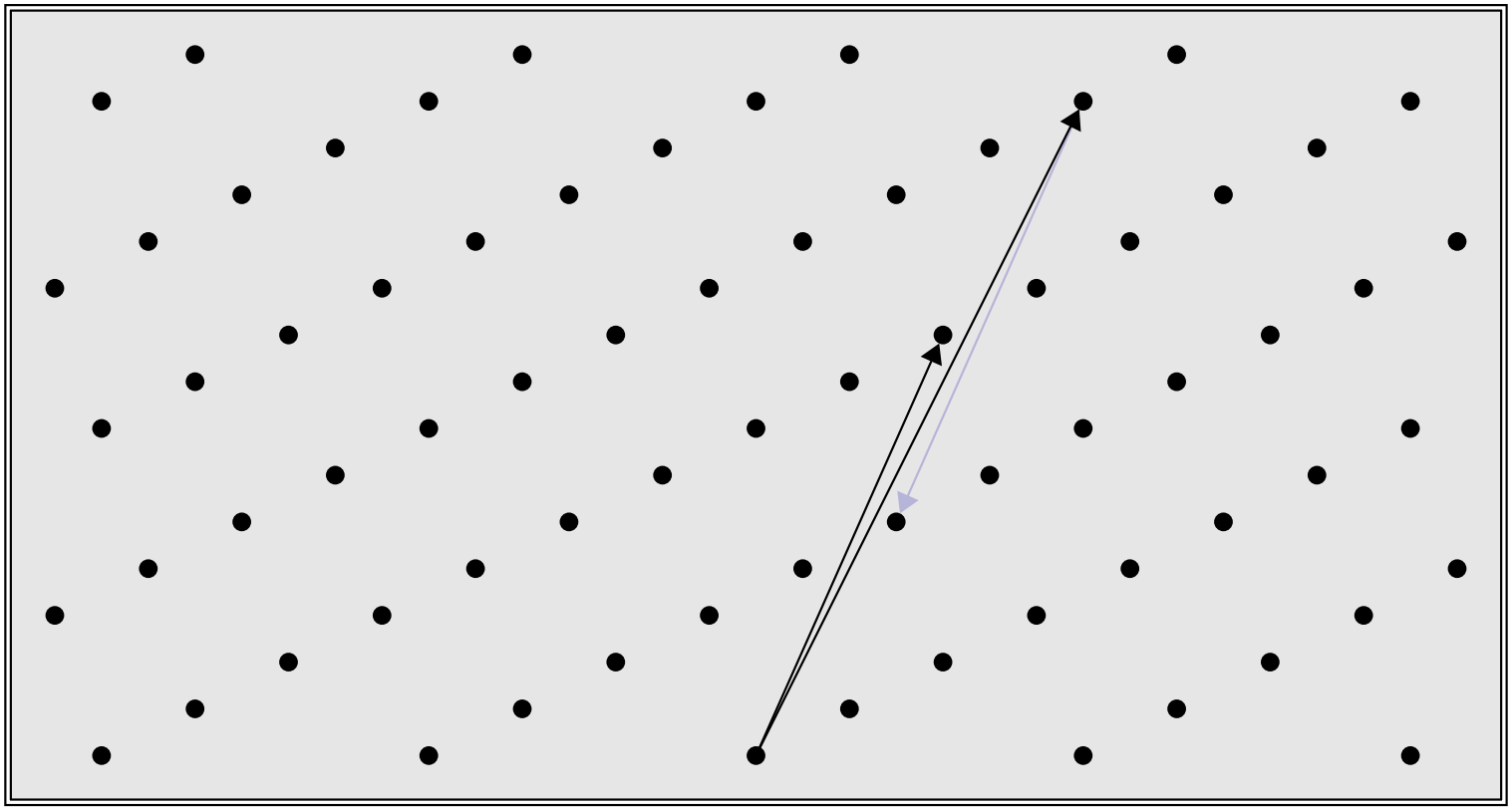


Lattice reduction in dimension 2

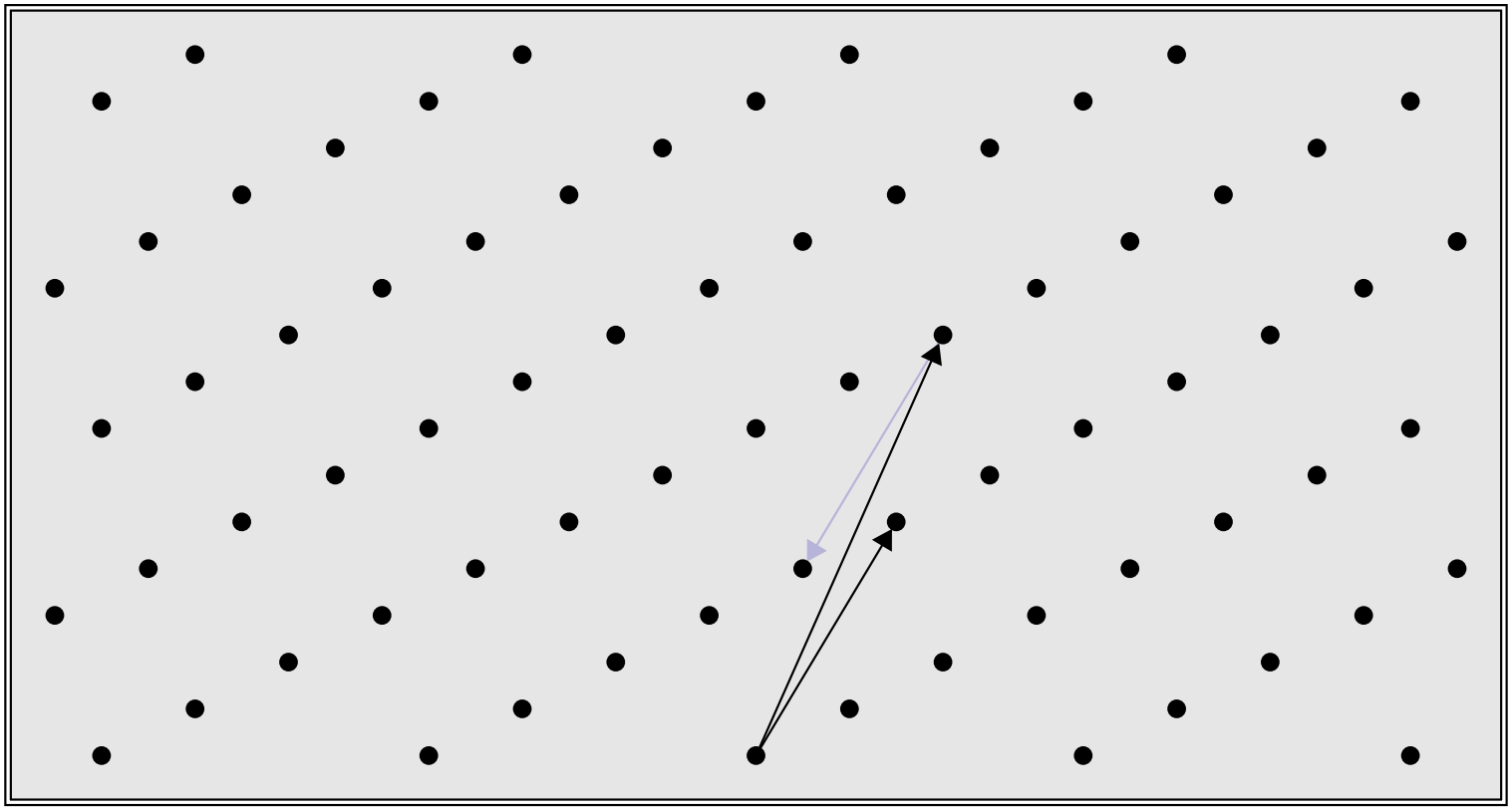
Gauss reduction algorithm

Require: Initial lattice basis $(\vec{u}, \vec{v})$
  if $\|\vec{u}\| < \|\vec{v}\|$ **then**
    Exchange $\vec{u}$ and $\vec{v}$
  end if
  repeat
    Minimize $\|\vec{u} - \lambda\vec{v}\|$, i.e., $\lambda \longleftarrow \left\lfloor \vec{u} \cdot \vec{v} / \|\vec{v}\|^2 \right\rceil$
    Let $\vec{u} \longleftarrow \vec{u} - \lambda\vec{v}$
    Swap $\vec{u}$ and $\vec{v}$
  until $\|\vec{u}\| \leq \|\vec{v}\|$
  Output $(\vec{u}, \vec{v})$ as reduced basis

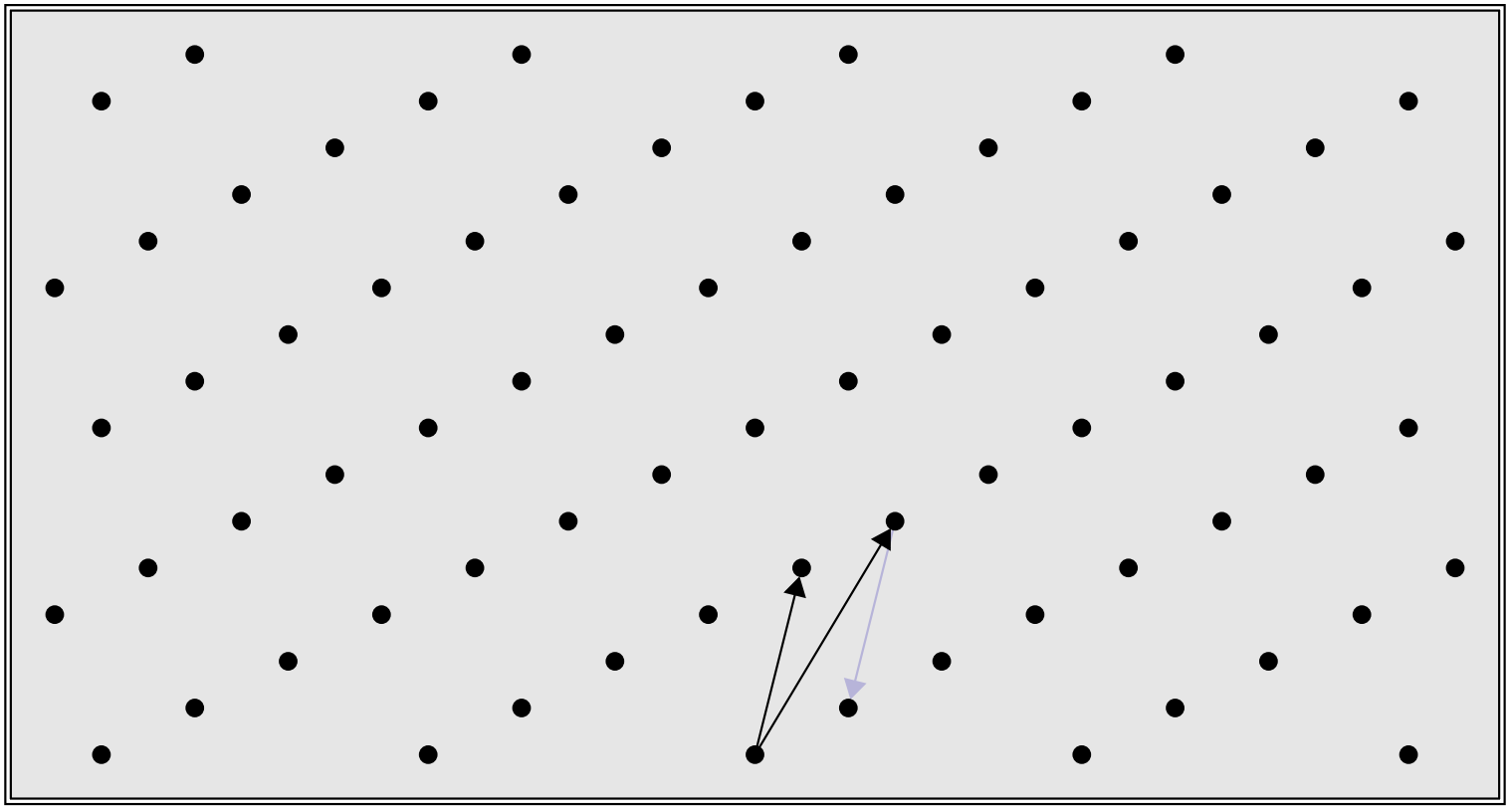
Gauss's reduction algorithm



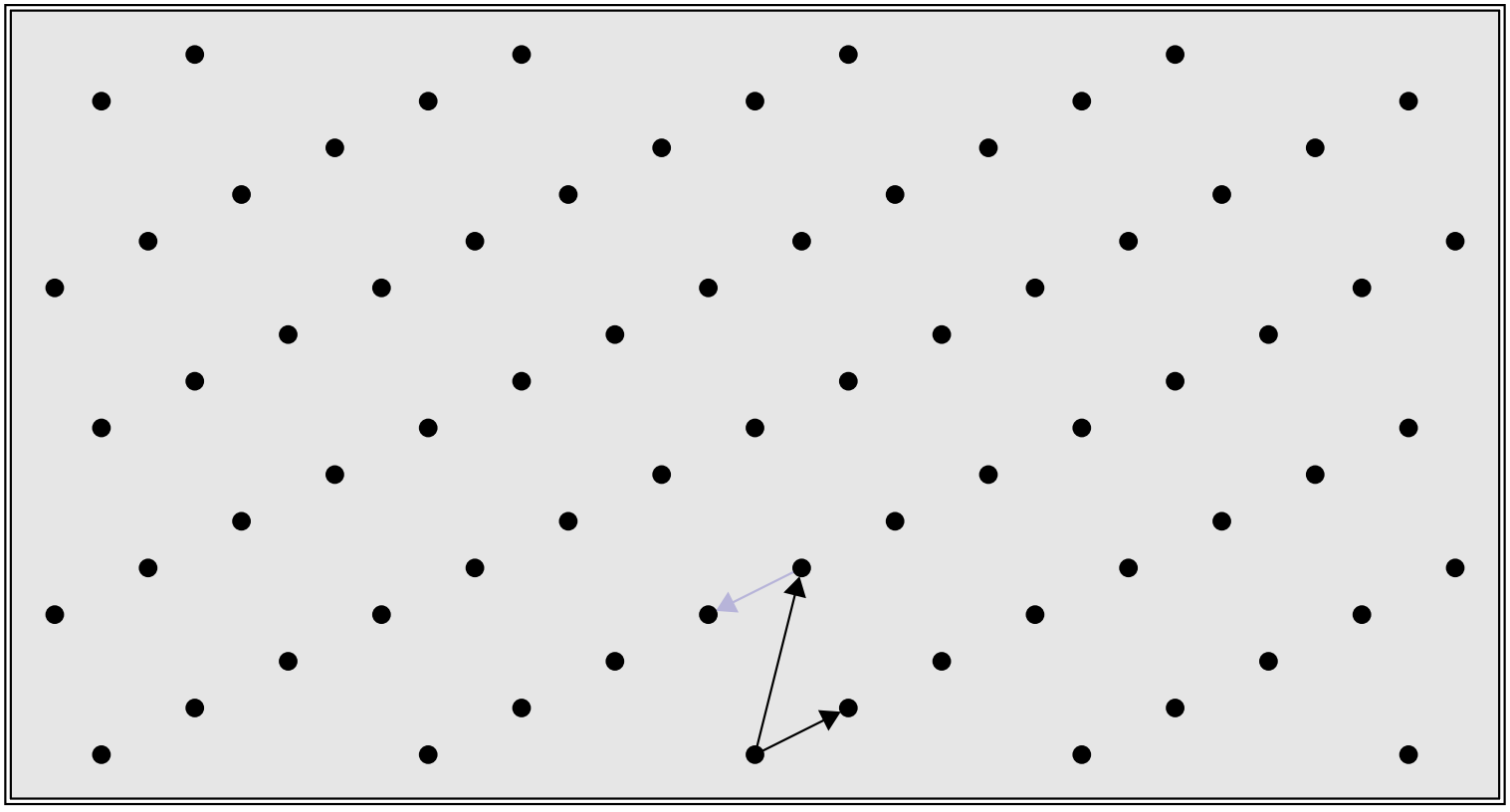
Gauss's reduction algorithm



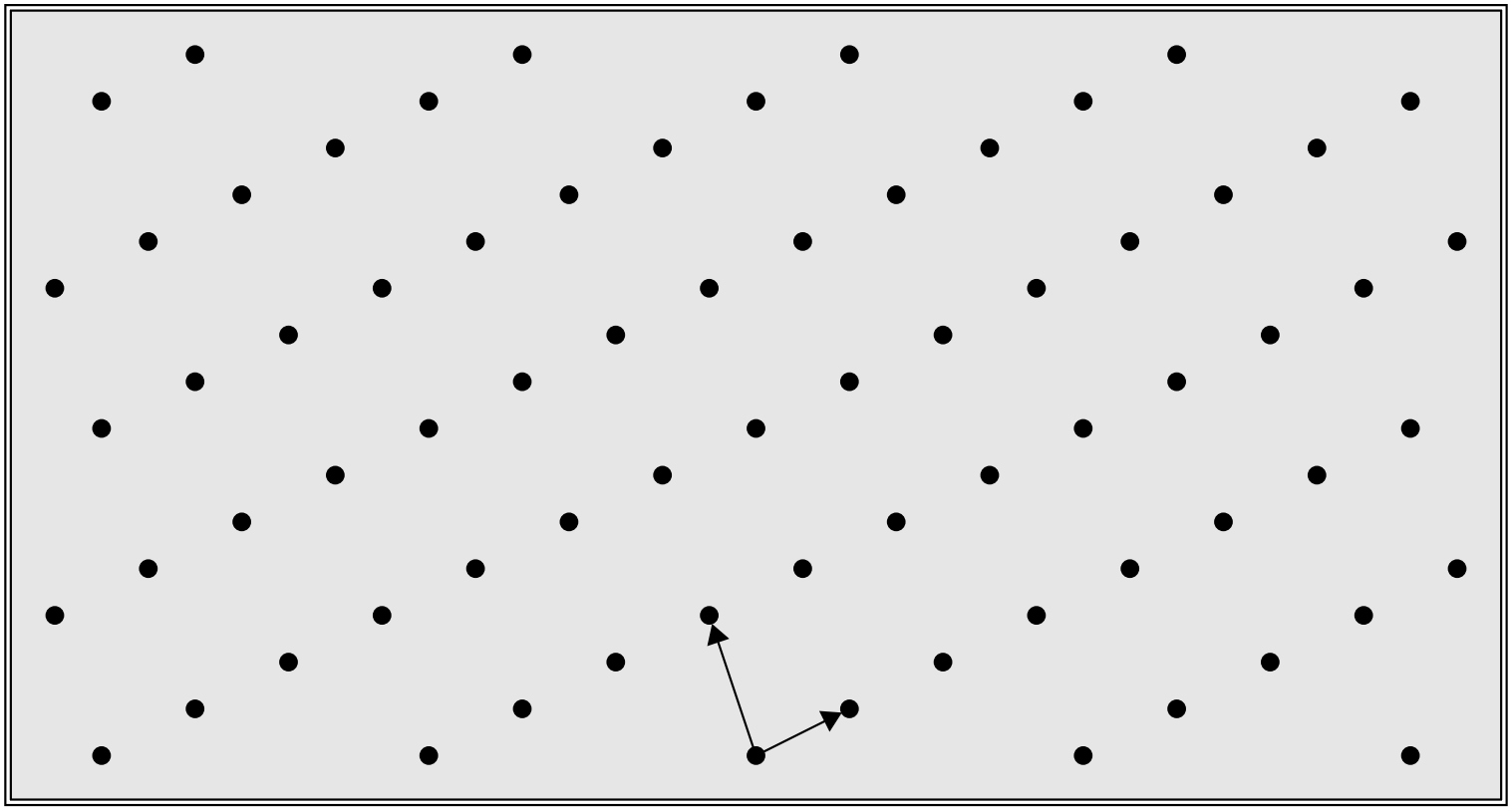
Gauss's reduction algorithm



Gauss's reduction algorithm



Gauss's reduction algorithm



Higher dimension: the LLL algorithm

- Invented by Lenstra, Lenstra and Lovász in 1982
- Polynomial time lattice reduction algorithm in arbitrary dimension
- Combines Gauss's algorithm and Gram-Schmidt orthogonalization

Reminder: the Gram-Schmidt Orthogonalization

- From family $\vec{b}$ compute orthogonal family $\vec{b}^*$ s.t. $\forall k \in [\![1, n]\!]$:
  - Spaces spanned by $(\vec{b}_1, \cdots, \vec{b}_k)$ and $(\vec{b}_1^*, \cdots, \vec{b}_k^*)$ are equal.
  - And $\vec{b}_k \cdot \vec{b}_k^* = \|\vec{b}_k^*\|^2$.
- Or $B = B^* \cdot M$ (M upper trg. with a diagonal of ones)

GSO algorithm

Require: Initial basis $(\vec{b}_1, \cdots, \vec{b}_n)$

Create a $n \times n$ matrix M and initialize it to identity.

for i from 1 to n **do**

$\vec{b}_i^* \longleftarrow \vec{b}_i$.

for i from 1 to n **do**

$$M_{i,j} \longleftarrow \frac{\vec{b}_j \cdot \vec{b}_j^*}{\|\vec{b}_j^*\|} \qquad \vec{b}_i^* \longleftarrow \vec{b}_i^* - M_{i,j}\, \vec{b}_j^*$$

end for

end for

Output $\vec{b}^*$ and the transformation matrix M

Key properties of LLL-reduced basis

- Enforces the following properties on the output basis:

$$\forall\, i < j \quad : \quad \left|\vec{b}_j \cdot \vec{b}_i^*\right| \leq \frac{\|\vec{b}_i^*\|^2}{2}$$

$$\forall\, i \quad : \quad \delta\|\vec{b}_i^*\|^2 \leq \left(\|\vec{b}_{i+1}^*\|^2 + \frac{(\vec{b}_{i+1} \cdot \vec{b}_i^*)^2}{\|b_i^*\|^2}\right)$$

- Implies (note: $1/4 < \delta \leq 1$):

$$(\delta - 1/4)\,\|\vec{b}_i^*\|^2 \leq \|\vec{b}_{i+1}^*\|^2$$

The LLL algorithm

LLL

Require: Basis $(\vec{b}_1, \cdots, \vec{b}_n)$
  Do GS Orthogonalization
  Let $k \longleftarrow 2$
  repeat
    Do **size reduction** of $\vec{b}_k$
    if $(\delta - M_{k,k-1}^2)\|\vec{b}_{k-1}^*\|^2 \leq \|\vec{b}_k^*\|^2$ **then**
      Let $k \longleftarrow k+1$
    else
      Exchange $\vec{b}_k$ and $\vec{b}_{k-1}$
      Update GSO
      Let $k \longleftarrow \max(2, k-1)$
    end if
  until $k > n$
  Output updated basis

Size reduction of $\vec{b}_k$

  for i from $k-1$ down to 1 **do**
    $\vec{b}_k \longleftarrow \vec{b}_k - \lfloor M_{k,i} \rceil \vec{b}_i$
    Update GSO
  end for
  return

Early cryptanalytic applications

Knapsacks a.k.a. Subset sums

- The subset-sum problem (or knapsack problem) is:
  - Given integers $a_1, \ldots, a_n$ and S
  - Find $\epsilon_1, \ldots, \epsilon_n$ with $0/1$ values such that:

$$S = \sum_{i=1}^{n} \epsilon_i a_i$$

- NP-hard problem
- Some cases are easy (e.g. $a_i = 2^{i-1}$)

Knapsack-based cryptosystems

- Main idea: Hide an easy knapsack in a hard-looking one

Merkle-Hellman cryptosystem

- Start from super-increasing knapsack where $a_i > \sum_{j=1}^{i-1} a_j$
- Choose $q > \sum_{i=1}^{n} a_i$ (prime for simplicity)
- Choose r a random integer modulo q
- Form new knapsack with $b_i = r a_{\pi(i)} \pmod{q}$
- **Encryption:** Compute $S = \sum_{i=1}^{n} \epsilon_i b_i$
- **Decryption:** Let $S_a = S\, r^{-1} \pmod{q}$ and solve easy knapsack

- Broken by Shamir at Crypto'82

Sketch of Shamir's attack

- Assume π is identity (or guess $\pi(1)$, $\pi(2)$, $\pi(3)$, $\pi(4)$)
- For simplicity, assume that b_1 and b_2 are coprime
- Let $c_3 = b_3/b_2 \pmod{b_1}$ and $c_4 = b_4/b_2 \pmod{b_1}$
- Form lattice (spanned by rows) :

$$\begin{pmatrix} 1 & c_3 & c_4 \\ 0 & b_1 & 0 \\ 0 & 0 & b_1 \end{pmatrix}$$

- Contains all vectors $(\lambda b_2, \lambda b_3, \lambda b_4)$ modulo b_1
- Remark that $a_1 b_i - a_i b_1 = u_i q$ and u_i "small"
- Yields short vector (u_2, u_3, u_4)

Sketch of Shamir's attack (continued)

- In particular: $a_1/q = u_i/b_i \pmod{b_1}$
- Let $\mu = u_i/b_i \pmod{b_1}$
- We can now decrypt with (mostly) equivalent key (μ, b_1)
- Indeed, $a_1 b_i - a_i b_1 = u_i q$ with u_i of the same order as a_i
- Thus, the knapsack u_i is super-increasing (in general)

Another approach to break Merkle-Hellman knapsack

- Since a_i is super-increasing, a_n has $2n$ bits
- So does q and all b_is
- Define density of a knapsack:

$$d = \frac{n}{\log_2(\max_i a_i)}$$

- As a general rule:

Low density $\Rightarrow$ Easy to solve

Basic low-density attack

- Lagarias-Odlyzko (1985)
- Consider the lattice generated by columns of:

$$\begin{pmatrix} Ka_1 & Ka_2 & \cdots & Ka_n & Ks \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

- With K large enough
  - LLL outputs short vector with 0 on the first line
- Short relation $\sum_{i=1}^{n} v_i a_i = v_{n+1} s$

Is it the correct $\{0, 1\}$ solution?

Integer points in n-spheres

Theorem (Mazo-Odlyzko)

For any $\alpha > 0$, there exists a constant $c(\alpha) > 0$ such that for all $\vec{x} \in \mathbb{R}^n$, the number $N(\vec{x}, n, \alpha)$ of integer points in the n-sphere centered at $\vec{x}$ satisfies:

$$N(\vec{x}, n, \alpha) \leq \exp(c(\alpha)\sqrt{n})\, N(\vec{0}, n, \alpha).$$

Moreover, there exists a (computable) constant $C(\alpha)$ such that:

$$N(\vec{0}, n, \alpha) \leq C(\alpha)^n.$$

Analysing low density attacks with reduction oracles

- Rand. knapsack a with **fixed** solution ϵ, sum s and density d:
  - Take random a_is in $[\![0, M-1]\!]$ with $M = \text{Nextprime}(\lfloor 2^{n/d} \rfloor)$
  - Let $s = \sum_{i=1}^{n} \epsilon_i \, a_i$
- Take short vector in the Lagarias-Odlyzko lattice. Observe:

$$\sum_{i=1}^{n} v_i \, a_i = v_{n+1} \, s$$

- Thus $\sum_{i=1}^{n} (v_i - \epsilon_i \, v_{n+1}) \, a_i \equiv 0 \pmod{M}$.
- Prob. $1/M$ *[except for multiples of the true solution]*

Basic low-density attack

- Probability of parasitic solution bounded by $(C(1/2)/2^{1/d})^n$
- Negligible when $d < 0.6463$
- Thus shortest lattice vector oracle $\Rightarrow$ correct solution
- Surprisingly:

Works well in practice!

- With LLL bounds, would need $d < O(1)/n$

Improved low-density attacks

- Consider the lattice generated by columns of:

$$\begin{pmatrix} Ka_1 & Ka_2 & \cdots & Ka_n & -Ks \\ 1 & 0 & \cdots & 0 & 1/2 \\ 0 & 1 & \cdots & 0 & 1/2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1/2 \\ 0 & 0 & \cdots & 0 & 1/2 \end{pmatrix}$$

- Improved bound $d < 0.9408$ (smaller sphere radius)

Warning: Closest vector oracle ?

A note of caution

- Despite these early success:
  - Lattice-reduction is hard
- In practice: Lattice-reduction works very well in moderate dimension
- In higher dimension, many problems appear:
  - Exponential gap between $\vec{b}_1$ and first minimum
  - Unstability problems
  - Running time and performance greatly depend on considered lattice

$\Rightarrow$ Would be nice to have attacks without oracles.

Reconstruction of small linear dependencies

- Let $(x_1, \cdots, x_n)$ be a sequence of reals
- We want to find small coefficients such that $\sum v_i x_i = 0$
- Consider the lattice generated by columns of:

$$\begin{pmatrix} \lfloor Ka_1 \rceil & \lfloor Ka_2 \rceil & \cdots & \lfloor Ka_{n-1} \rceil & \lfloor Ka_n \rceil \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Typical example

- Let $x = -3.5795612178451722\cdots$
- Form the lattice:

$$\begin{pmatrix} 100000 & -357956 & 1281326 & -4586584 & 16417959 & -58769091 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- LLL find a short vector $(-2, 7, 1, 0, -2, 3, 1)$
- I.e. the polynomial: $x^5 + 3\,x^4 - 2\,x^3 + x + 7$

Contructing polynomials for discrete log NFS

- Variation to find a polynomial P s.t. $P(\theta) = 0 \pmod{p}$
- Consider the lattice generated by columns of:

$$\begin{pmatrix} K & K\theta & \cdots & K\theta^n & Kp \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

- Any vector starting with a 0 works

Coppersmith attack

- Modular version, solve polynomial equation:

$$f(x) = 0 \pmod{N}.$$

  Easy when factorization of N is known. Hard in general.

- Bivariate version, find integral roots of:

$$f(x, y) = 0.$$

  Diophantine equations. Hard in general.

Variant (for simplified presentation)

- Search rational solutions
- Equivalently, consider homogeneous polynomials
- Modular version, solve polynomial equation:

$$f(x_0, x_1) = 0 \pmod{N}.$$

- Bivariate version, find integral roots of:

$$f(x_0, x_1, y_0, y_1) = 0.$$

  Homogeneous separately in x and y.

A simple case (Howgrave-Graham's variation)

- Search small solutions of:

$$f(x_0, x_1) = a\,x_0^2 + b\,x_0\,x_1 + c\,x_1^2 = 0 \pmod{N}.$$

  W.l.o.g, we may assume $c = 1$.
- Fix two parameters, D and t
- Consider homogeneous polynomials of degree D with root (x_0, x_1) modulo N^t
- Obtained by linearly combining:

$$x_0^{D-2i}\,f(x_0, x_1)^i\,N^{\max(0,t-i)} \quad \text{and}$$
$$x_0^{D-2i-1}\,x_1\,f(x_0, x_1)^i\,N^{\max(0,t-i)}$$

A simple case

- Use monomial ordering with $x_1 > x_0$
- Head monomial in

$$x_0^{D-2i-\theta}\, x_1^{\theta}\, f(x_0, x_1)^i\, N^{\max(0,t-i)}$$

is $x_1^{2i+\theta}\, x_0^{D-2i-\theta}$ and has coefficient $N^{\max(0,t-i)}$

Interpret polynomials as lattice points

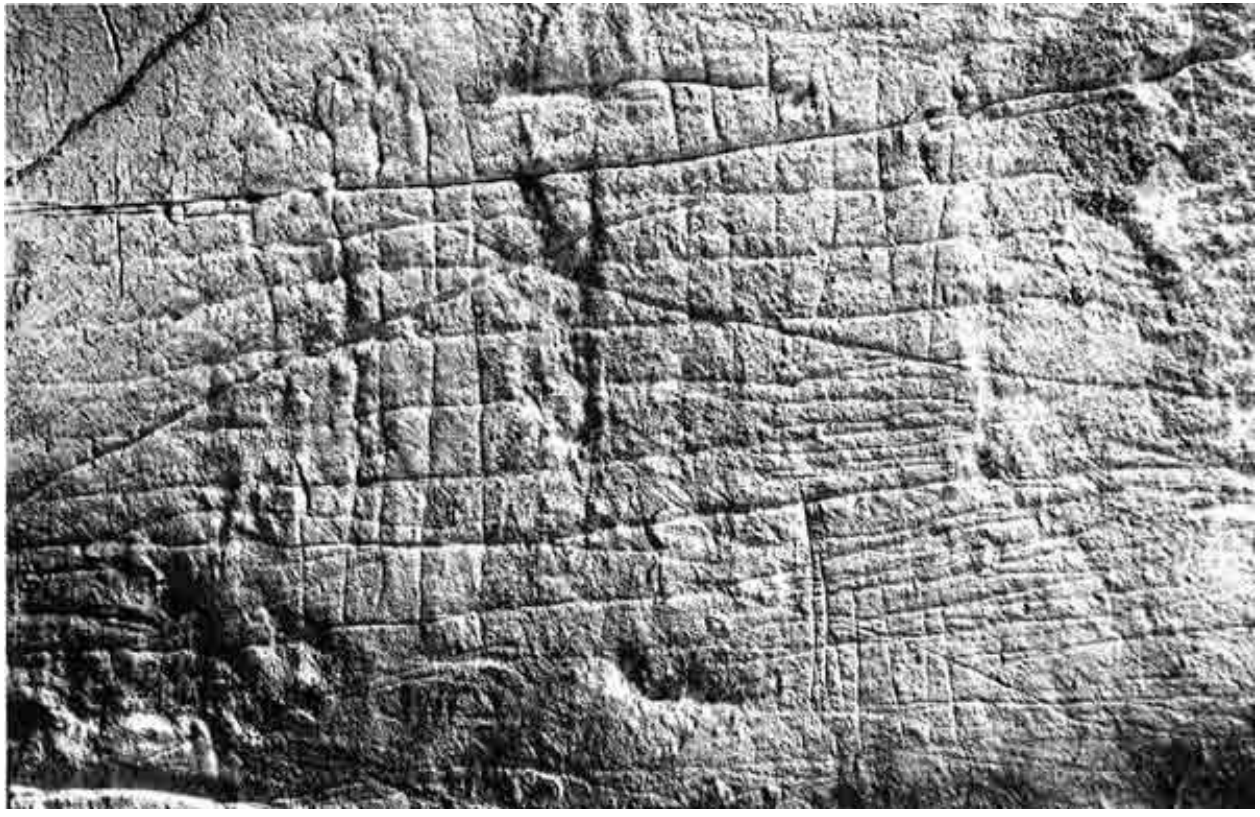
$$([x_0^D], [x_0^{D-1}x_1], \cdots, [x_0 x_1^{D-1}], [x_1^D])$$

End of the simple case

- As a consequence, get polynomial F with $F(x_0, x_1) = 0$ over $\mathbb{Z}$
- Dehomogenizing, we find $F_a(x_0/x_1) = 0$
- Solve over $\mathbb{R}$
- Recover x_0 and x_1 from root r using continued fractions

f of degree d $\Rightarrow$ Works up to $N^{1/2d}$ bound on x_0 and x_1

Conclusion



Conclusion