Lattices

Brice Minaud

email: brice.minaud@ens.fr
website: www.di.ens.fr/brice.minaud/init-crypto.html

Initiation à la Cryptologie, ENS/MPRI, 2019-2020

Meta information

Exam: Monday, May 25, 2pm to Wednesday 27, 5pm.

Register here:

https://www.di.ens.fr/david.pointcheval/cours.html

All other info for this course, including past lectures/TAs:

https://www.di.ens.fr/brice.minaud/init-crypto.html

(This time there is no difference with last week.)

Reminder: hard problems in post-quantum world

Post-quantum candidate hard problems:

- Lattices.
- Code-based crypto.
- Isogenies.
- Symmetric crytpo (→ signatures).
- Multivariate crypto.

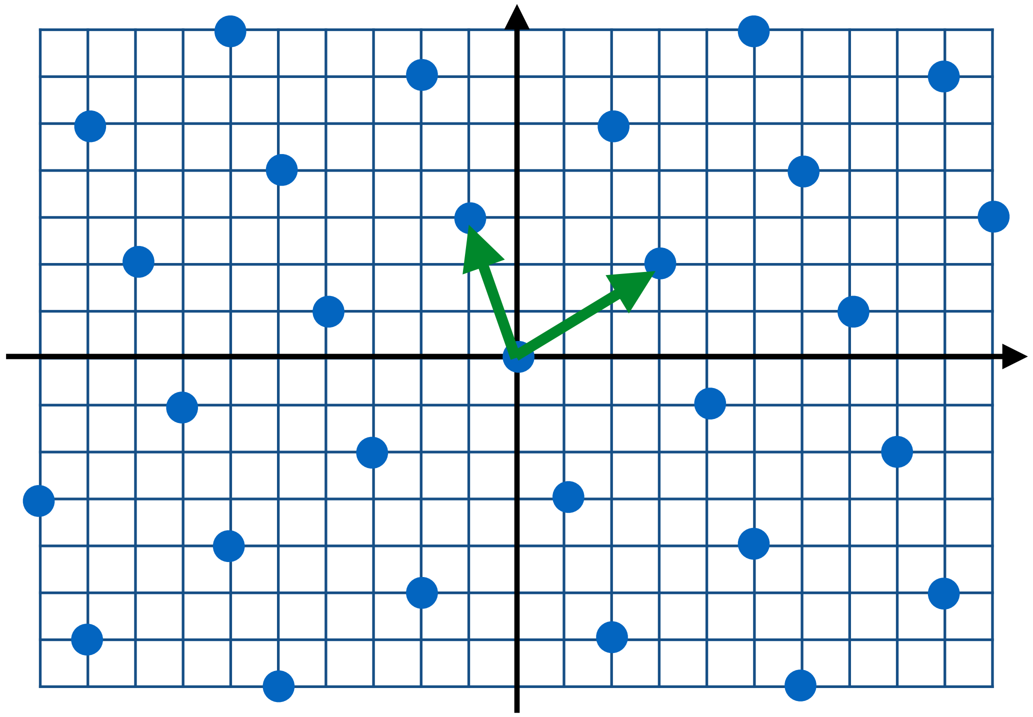
Number Theory

Lattices, codes,…
(conjectured)

Lattices are the mainstream candidate. Other PQ approaches for Public-Key crypto "only" motivated by PQ. Lattice-based crypto stands on its own:

- Simplicity (of schemes, not analysis).
- Security from worst-case hardness.
- Very expressive/verstatile, much beyond PKE/sig.

Lattices



Lattices

Lattice. A lattice $\mathscr{L}$ is:

- An additive subgroup of $\mathbb{R}^n$.
- Discrete (not dense).

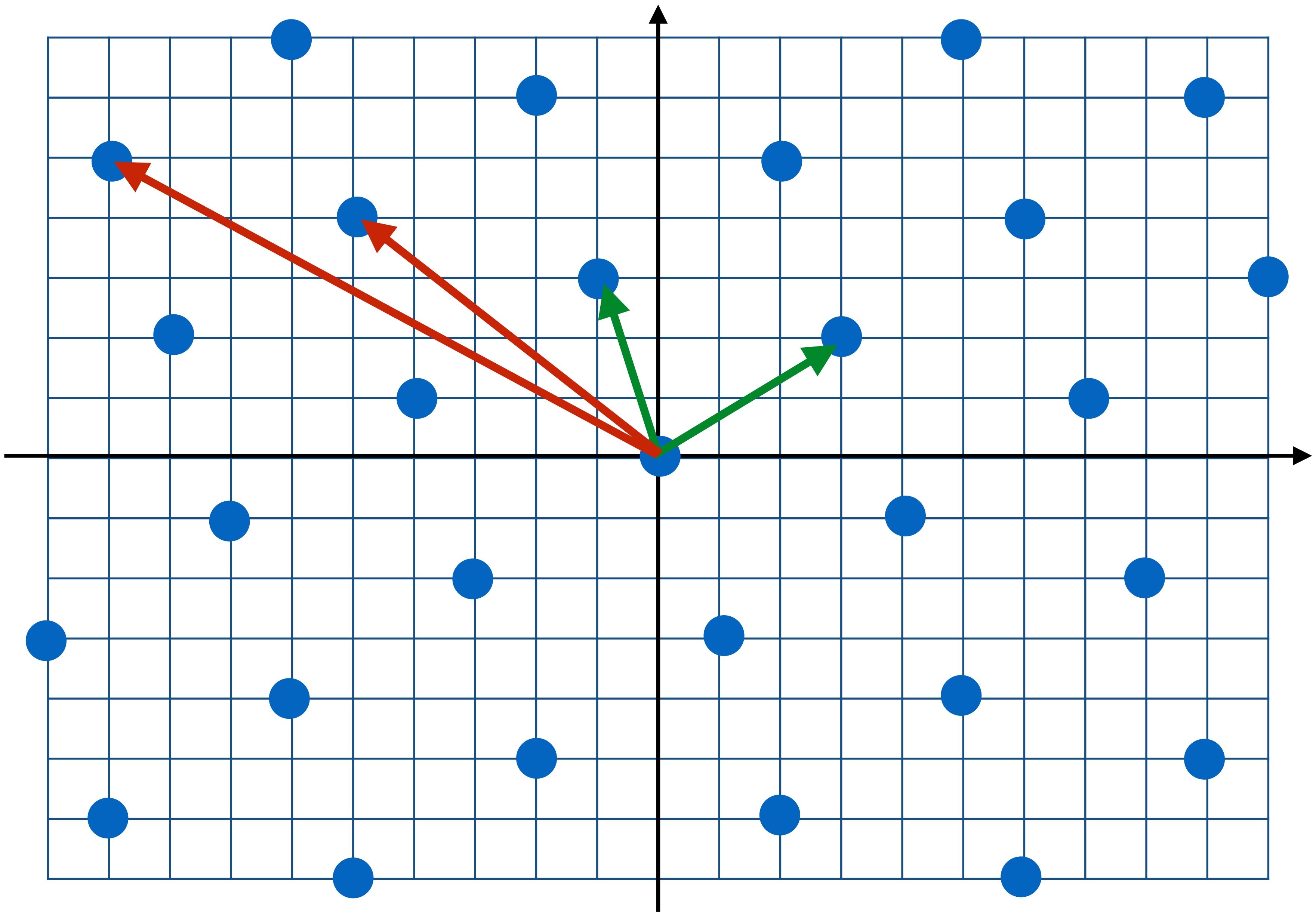
In practice, in crypto, $\mathscr{L}$ often:

- Spans $\mathbb{R}^n$, a.k.a. "full-rank".
- Typically $\subseteq \mathbb{Z}^n$.
- Often "q-ary": all qe_i = (0,…,0,q,0,…,0)'s are in $\mathscr{L}$. That is, the lattice wraps around mod q. Can be regarded as in $\mathbb{Z}_q^n$.

Concretely, $\mathscr{L}$ can be defined by a basis $B \in \mathbb{Z}^{n \times n}$:

$$\mathscr{L} = B\mathbb{Z}^n$$

In pictures



Basis B.

Basis B'.

Dual lattice

Dual lattice. The **dual** $\mathscr{L}^*$ of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$ is:

$$\mathscr{L}^* = \{x \in \mathbb{R}^n : \forall\, y \in \mathscr{L},\ {}^t x y \in \mathbb{Z}\}$$

Properties of the dual:

- It is a lattice.
- It characterizes the lattice $\mathscr{L}$: $\mathscr{L}^{**} = \mathscr{L}$.
- If B is a basis of $\mathscr{L}$, $({}^t B)^{-1}$ is a basis of $\mathscr{L}^*$.

Hermite Normal Form

A lattice can be charaterized by a basis in **Hermite Normal Form**.

HNF basis is unique and easy to compute from any basis → "neutral" description of the lattice.

Hermite Normal Form. A basis B ∈ $\mathbb{Z}^{n x n}$ of a (full-rank) lattice is HNF iff:

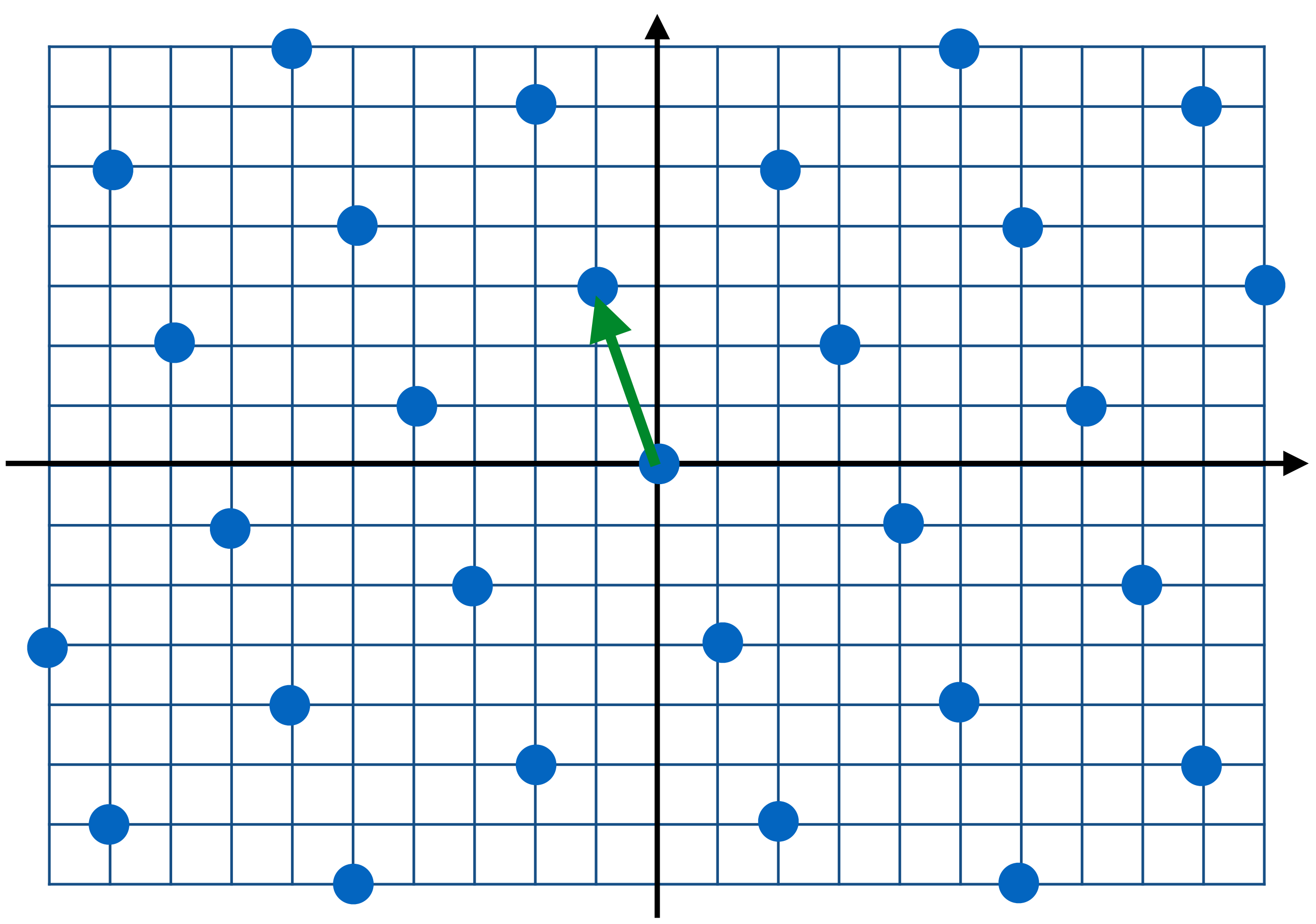
- It is upper triangular, with > 0 diagonal elements.
- Elements to the right of a diagonal element $m_{i,i}$ are ≥0 and < $m_{i,i}$.

Hard problems in lattices

Define the usual ℓ^2 norm on $\mathbb{R}^n$.

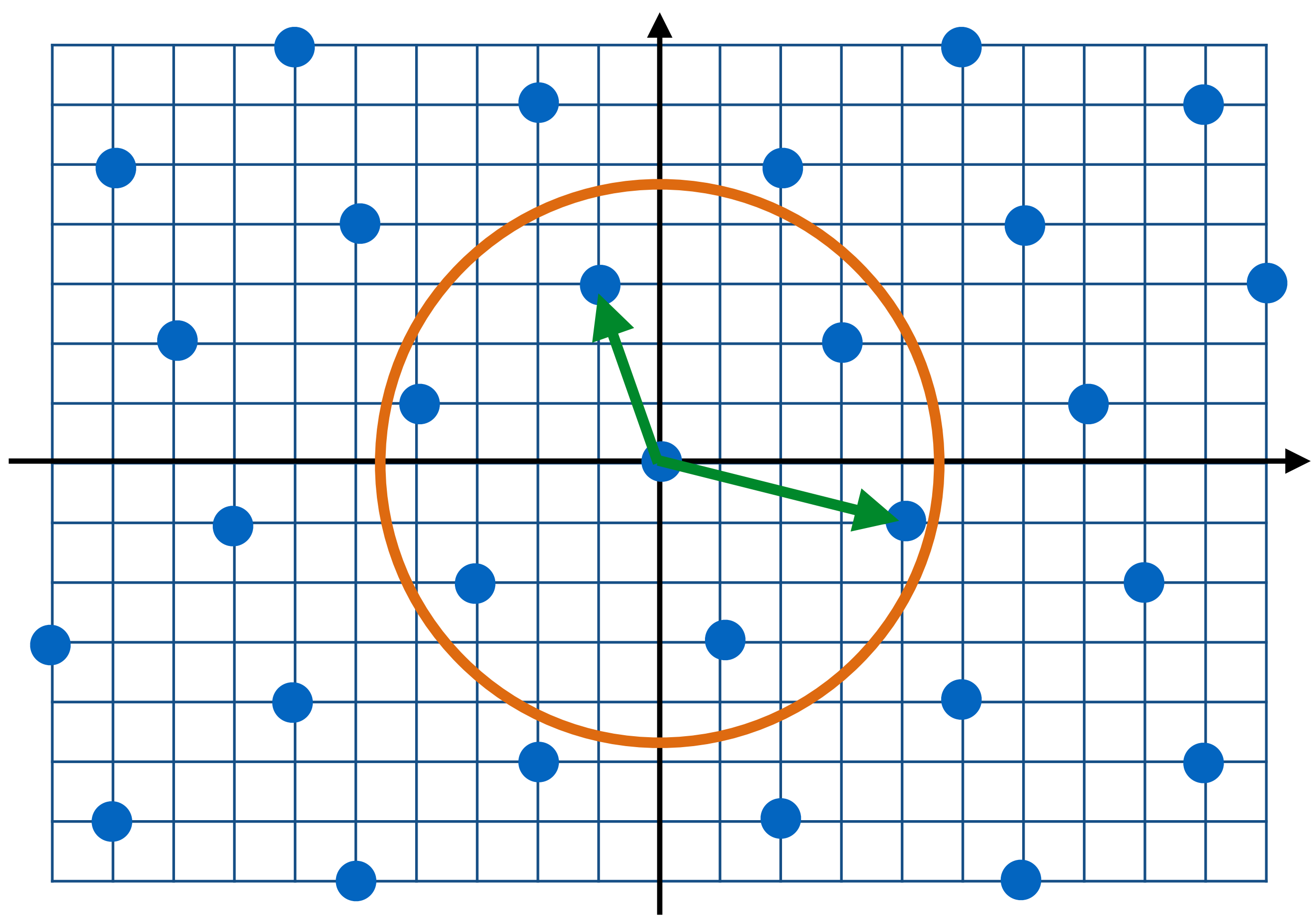
Define $\lambda_i(\mathscr{L})$ to be the smallest vector independent from $\lambda_1(\mathscr{L}), \ldots, \lambda_{i-1}(\mathscr{L})$.

Shortest Vector Problem (SVP). Given a basis B of a lattice $\mathscr{L}$, find the smallest non-zero lattice vector. I.e., find x $\in \mathscr{L}$ s.t. $||x|| = \lambda_1(\mathscr{L})$.



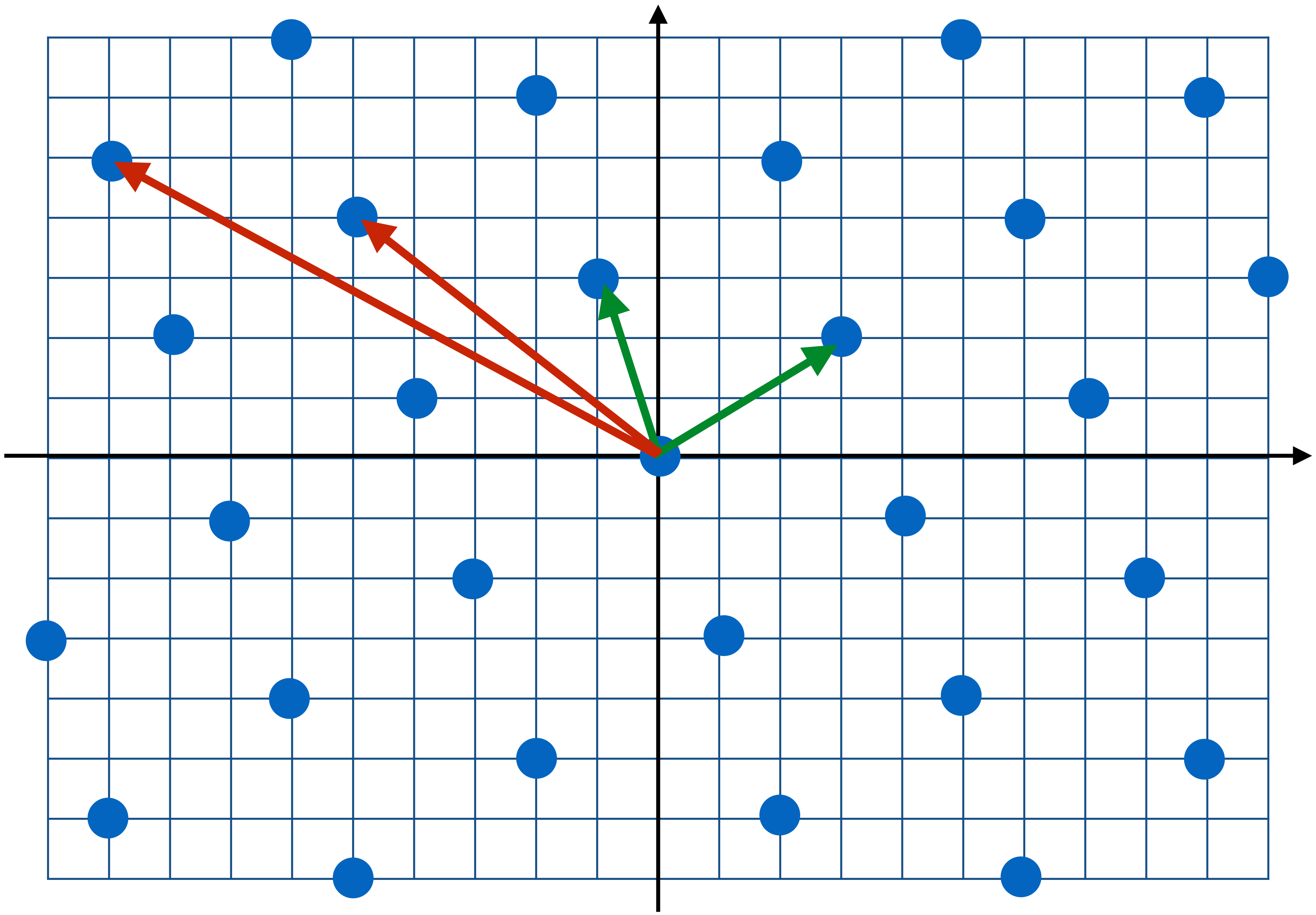
Hard problems in lattices

Shortest Vector Problem (SVP$_\gamma$). Given a basis B of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, find a vector x of norm $\leq \gamma(n) \cdot \lambda_1(\mathscr{L})$.



Decisional Shortest Vector Problem (GapSVP$_\gamma$). Given a basis B of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, decide if $\lambda_1(\mathscr{L}) \leq 1$ or $\lambda_1(\mathscr{L}) \geq \gamma(n)$.

In pictures

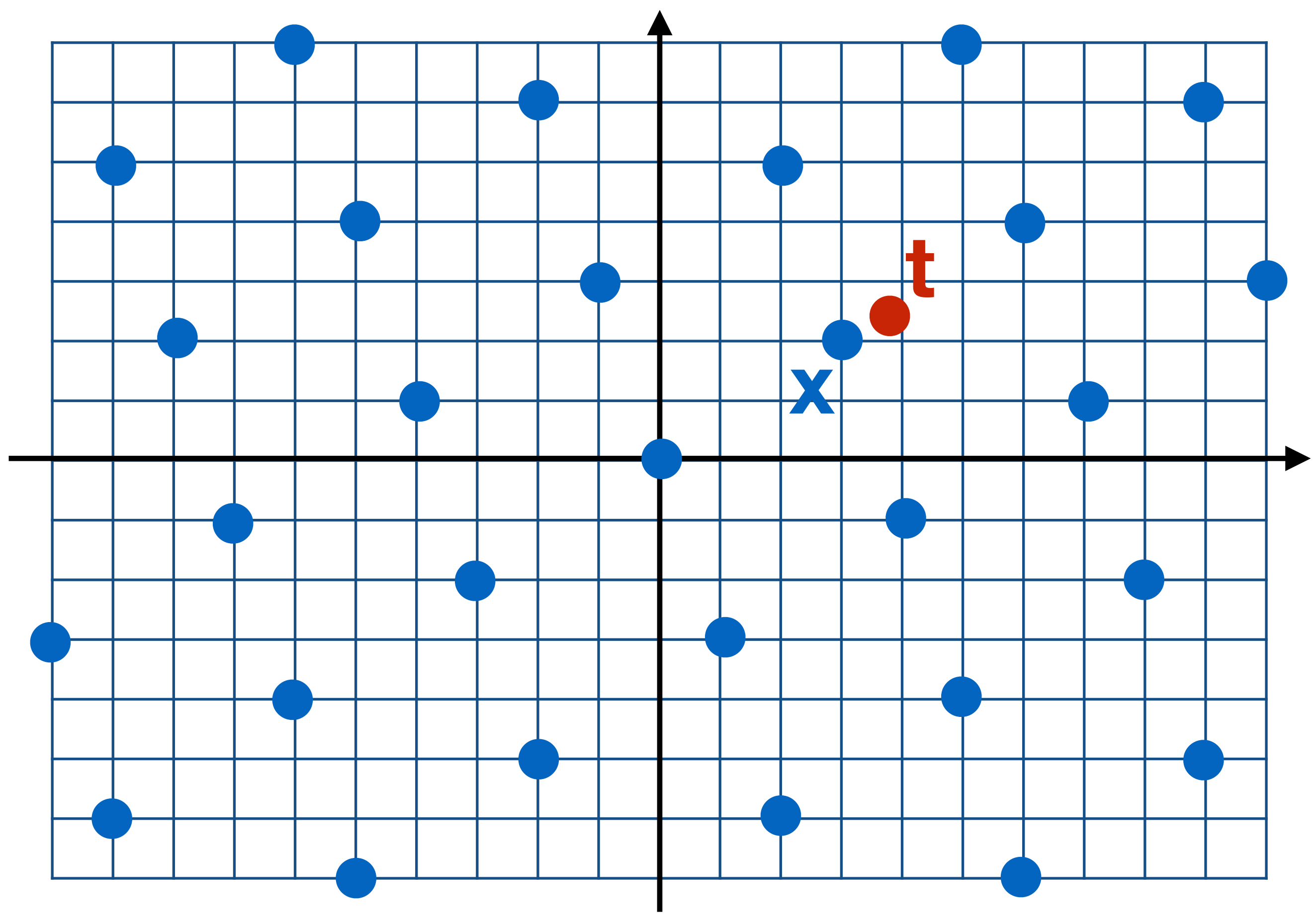


Good basis.

Bad basis.

Hard problems in lattices

Bounded Distance Decoding (BDD$_\gamma$). Given a basis B of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$ and t$\in\mathbb{R}^n$, with the promise: ∃ x$\in \mathscr{L}$, $||t - x|| < \lambda_1(\mathscr{L})/(2\gamma(n))$, find x (necessarily unique for $\gamma \geq 1$).

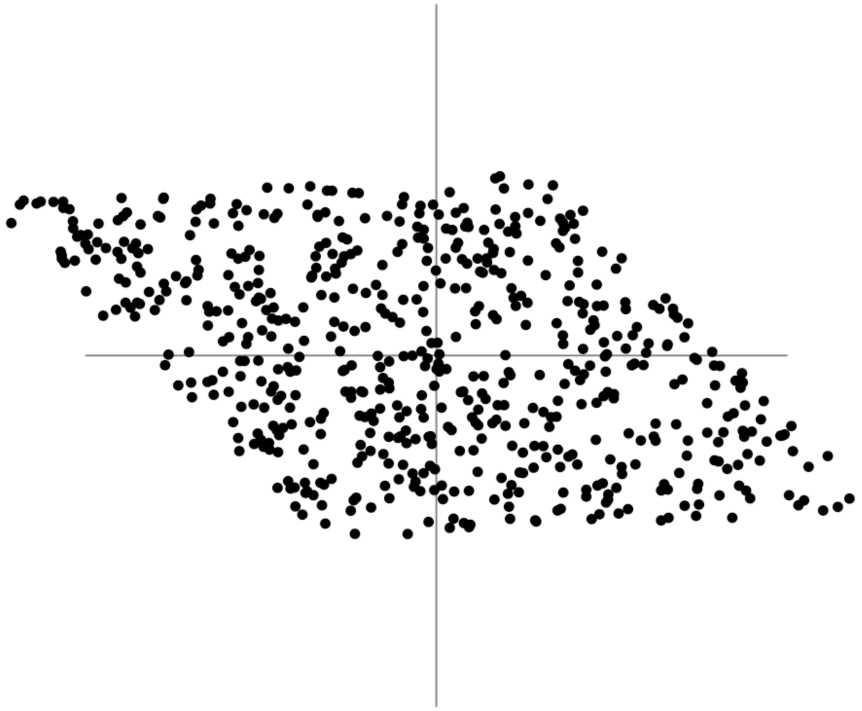


How hard are these problems?

- Deep and well-studied area → confidence in hardness.
- No known significant quantum speedup.
- Worst-case to average-case reduction.
- However, not (believed to be) NP-hard.

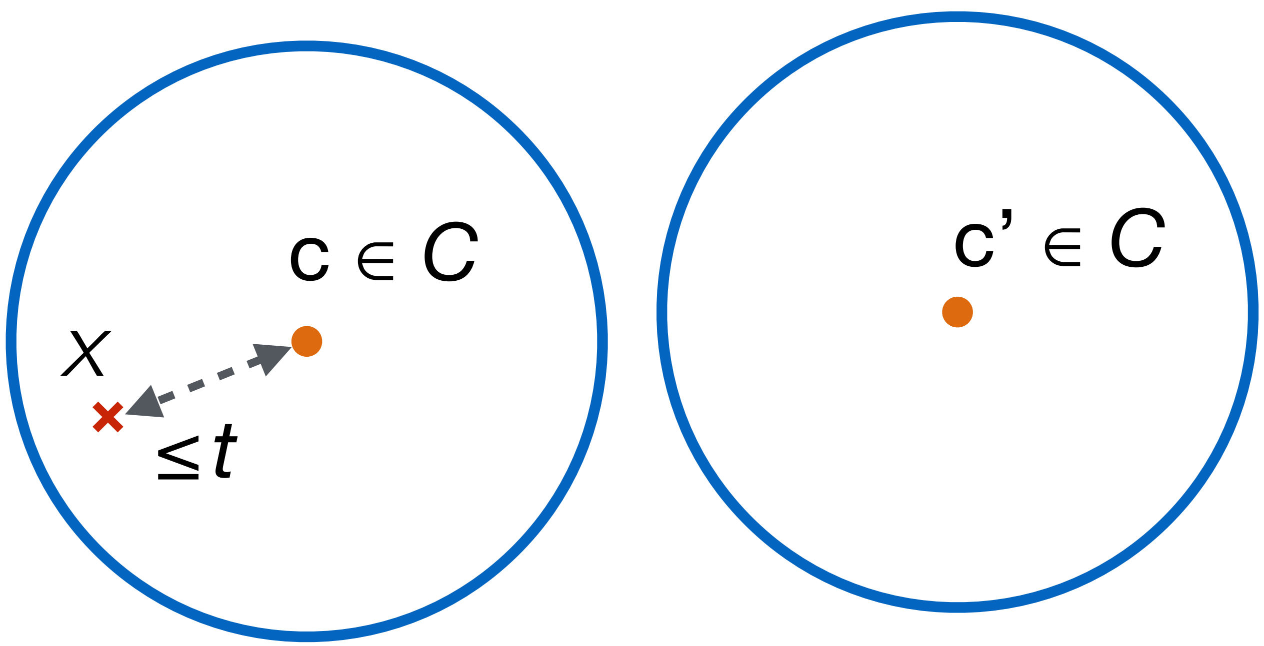
  For typical choice in crypto of $\gamma \geq \in \mathrm{Poly}(n)$ with $\gamma \geq \sqrt{n}$, GapSVP is in NP∩coNP.

Crypto from lattices



Recall code-based crypto…

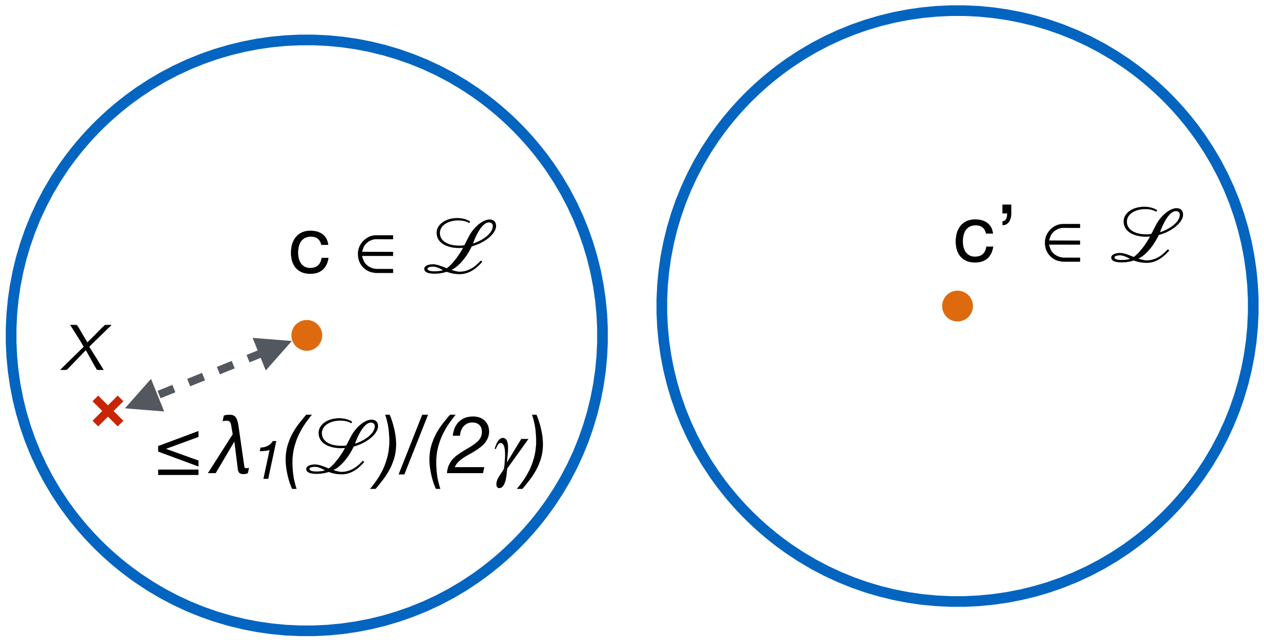
Problem: given a generator matrix G (i.e. a basis of C) and some x such that dist(x-c) $\leq t$ for some c in C, find c.



- For a random linear code, this is a **hard problem**!
- Except if you have a trapdoor (the code is secretly a "permutation" of an efficiently decodable code).

Now with lattices…

Problem: given a random lattice in $\mathbb{Z}_q$ (given as HNF of a uniform matrix) and some x such that dist(x-$\mathscr{L}$) ≤ $\lambda_1(\mathscr{L})/2\gamma$, find *c*.



- This is **BDD$_\gamma$**! It is a **hard problem**.
- Except if you have a trapdoor: namely, a good base of the lattice. You can then apply Babai's rounding algorithm.

The McEliece cryptosystem

Robert McEliece, 1978.

Pick a binary t-correcting Goppa code with generator matrix G.

Public key: $G' = S \cdot G \cdot P$, where S is a random invertible matrix, and P is a random permutation matrix.

Secret key: S, G, P.

Encrypt: encode a message m into the code C' (generated by G'), pick a random error vector e of weight t. The ciphertext c is:

$$c = m + e$$

Decrypt: given a ciphertext c, decode c using knowledge of the equivalence between C and C' (via S, P).

The GGH cryptosystem

Golreich, Goldwasser, Halevi 1997.

Pick a good basis G of some lattice L in $\mathbb{Z}_q$.

Public key: Hermite Normal Form B of G.

Secret key: G.

Encrypt: encode a message m into the lattice L (generated by B), pick a small enough random error vector e. The ciphertext c is:

$$c = m + e$$

Decrypt: given a ciphertext c, retrieve closest lattice point m using knowledge of the good basis G (using Babai's rounding algorithm).

The GGH cryptosystem

- Warning: Like RSA or basic McEliece, this is actually a **trapdoor permutation**. It is not a PKE: not IND-CCA secure (why?).
- Some care is needed regarding how the message is encoded into the lattice.
- In theory: **No reduction** → “heuristic” security.
- In practice: impossibly large parameters.

GGH signatures

Golreich, Goldwasser, Halevi 1997.

Pick a good basis G of some lattice L in $\mathbb{Z}_q$.

Public key: Hermite Normal Form B of G.

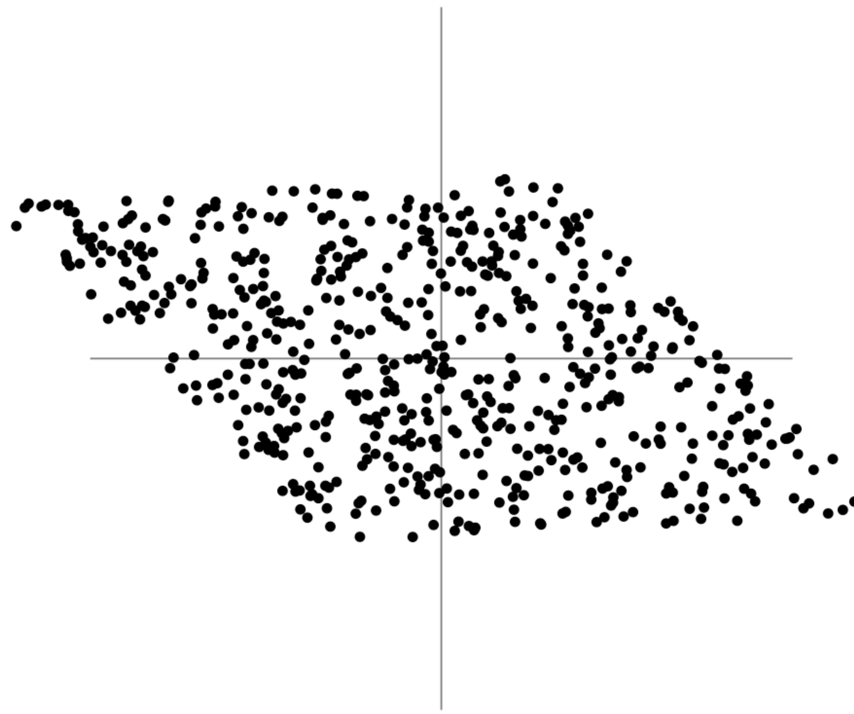
Secret key: G.

Sign: encode a message m as a point in $\mathbb{Z}_q$. The signature of m is the closest lattice point x (computed using G).

Verify: check that the signature x is close enough to m.

GGH signatures

- This time, similarities to Niederreiter signatures in codes.
- Again, **no reduction** → "heuristic" security.
- In fact, broken asymptotically and in practice! Nguyen-Regev '06.



- Idea: the value *x-m* is uniformly distributed in the fundamental parallelipiped $G \cdot [-1/2,1/2]^n$. Yields a learning problem: the Hidden Parallelipiped Problem.

Modern approach, part I

SIS: short integer solution

Short Integer Solution (SIS)

Ajtai '96 (the foundational article of Lattice-based crypto).

Say I have $m > n$ vectors a_i in $\mathbb{Z}_q^n$.

Problem: find **short** $x = (x_1,\ldots,x_m)$ in $\mathbb{Z}_q^m$ such that $\sum x_i a_i = 0$.
Here, **short** means of small norm: $\|x\| \leq \beta$.

- The crucial point is the norm constraint β. Otherwise this is just a linear system.
- Typically, Euclidian norm, with representatives in $[-q/2, q/2]$.
- Solution must exist as long as there are at least q^n vectors of norm $\leq \beta/\sqrt{2}$, due to collisions. E.g. $\beta > \sqrt{n \log q}$ and $m \geq n \log q$.

SIS and lattices

Equivalent formulation:

SIS problem. Given a uniform matrix $A \in \mathbb{Z}_q^{n \times m}$, find $x \in \mathbb{Z}_q^m$ with and $||x|| \leq \beta$ such that $Ax = 0$.

For A as above, define $\mathscr{L}^{\perp}(A) = \{x \in \mathbb{Z}_q^m : Ax = 0\}$(in $\mathbb{Z}_q$).

This is a (q-ary) lattice!

SIS = finding a short vector in $\mathscr{L}^{\perp}(A)$.

Better! Ajtai '96: Solving SIS (for uniformly random A) implies solving $\text{GapSVP}_{\beta\sqrt{n}}$ in dimension n for **any** lattice!

→ "Worst-case to average-case" reduction. Note m irrelevant.

(Cryptographic) hash function

Hash function H: $\{0,1\}^* \rightarrow \{0,1\}^n$.

Preimage resistance: for uniform $y \in \{0,1\}^n$, hard to find x such that $H(x) = y$.

Collision resistance: hard to find $x \neq y \in \{0,1\}^*$ such that $H(x) = H(y)$.

Note: collision is ill-defined for a single hash function. (why?)

→ To formally define hash functions, usually assume they are a *family* of functions. Parametrized by a "key".

(See also Random Oracle Model.)

(Cryptographic) hash function

In theory, collision-resistance ⇒ preimage resistance.

Argument: if the hash function is "compressing" enough, whp the preimage computed by a preimage algorithm, on input H(x), will be distinct from x. (Because most points will have many preimages.)

In practice, preimage resistance should cost 2^n, while collision resistance should cost $2^{n/2}$. → Previous reduction is not so relevant.

Right now we are more in the world of theory, so we'll only care about collision resistance.

Ajtai's hash function

Pick random $A \in \mathbb{Z}_q^{n \times m}$. Define:

$$H_A : \{0,1\}^m \to \mathbb{Z}_q^n$$
$$x \mapsto Ax$$

Finding a collision for random *A* yields a SIS solution with $\beta = \sqrt{m}$.

Indeed, $H_A(x) = H_A(x)$ yields $A(y\text{-}x) = 0$ with $y\text{-}x \in \{-1,0,1\}^m$.

Example: $q = n^2$, $m = 2n \log q$ (compression factor 2), need roughly $n \sim 100$, $mn \sim 100000$…

Modern approach, part II

LWE: learning with errors

Learning Parity with Noise (LPN)

Say I have $m > n$ vectors a_i in $\mathbb{Z}_2^n$.
I am given $a_i \cdot s + e_i$ (scalar product) for some secret s, $e_i \in \mathbb{Z}_2$ drawn from Bernoulli distribution B(η) (i.e. $\Pr(e_i = 1) = \eta$).

Problem: find s.

Oracle $O_\$$: returns (a,b) for a uniform in $\mathbb{Z}_2^n$, b uniform in $\mathbb{Z}_2$.
Oracle O_s: returns $(a, a \cdot s + e)$ for a uniform in $\mathbb{Z}_2^n$, e drawn from B(η).

LPN problem. Let $s \in \mathbb{Z}_2^n$ be drawn uniformly at random. Given access to either $O_\$$ or O_s, distinguish between the two.

LPN problem (bounded samples). Let $A \in \mathbb{Z}_2^{m \times n}$ and $b, s \in \mathbb{Z}_2^n$ be drawn uniformly at random, and $e \in \mathbb{Z}_2^m$ drawn according to B(η).
Distinguish between $(A, As + e)$, and (A, b).

Learning Parity with Noise (LPN)

- Famous problem in learning theory.
- Trivial without the noise.
- Believed to be very hard, even given unbounded samples. Best algorithm slightly sub-exponential: Blum-Kalai-Wasserman 2003. Complexity roughly $2^{n/\log n}$ in time and #queries.
- For bounded samples, same as decoding a random linear code.

Secret-key encryption using LPN

Attempt #1.

Pick a secret *s* uniformly in $\mathbb{Z}_2^n$.

Secret key: *s*.

Encrypt: to encrypt one bit *b*: give *m* samples from $O_\$$ if b=0, *m* samples from O_s if b=1.

Decrypt: use *s* to distinguish the two oracles.

Secret-key encryption using LPN

Attempt #2.

Pick a secret s uniformly in $\mathbb{Z}_2^n$.

Secret key: s.

Encrypt: to encrypt one bit b: give m samples from $(a, a \cdot s + b + e)$.

Decrypt: compute $a \cdot s$ to retrieve $b+e$, determine e by majority vote.

Secret-key encryption using LPN

Attempt #3.

Pick a secret S uniformly in $\mathbb{Z}_2^{m \times n}$.

Secret key: S.

Encrypt: to encrypt message m: (a,Sa+C(m)+e) where C(·) encodes the message into $\mathbb{Z}_2^m$ with error correction.

Decrypt: use S to retrieve C(m)+e, use error correction to remove e.

Additional tweaks: LPN-C cryptosystem (Gilbert et al. ’08).

Learning with Errors (LWE)

Regev '05. Milestone result.

Pick *s* uniformly in $\mathbb{Z}_q^n$.

Oracle $O_\$$: returns (*a*,*b*) for *a* uniform in $\mathbb{Z}_q^n$, *b* uniform in $\mathbb{Z}_q$.

Oracle O_s: returns (*a*,*a*·*s*+*e*) for *a* uniform in $\mathbb{Z}_q^n$, *e* drawn from χ.

Typically, χ is a discrete Gaussian distribution with std deviation αq.

LWE. Let $s \in \mathbb{Z}_q^n$ be drawn uniformly at random. Given access to either $O_\$$ or O_s, distinguish between the two.

LWE (bounded samples). Let $A \in \mathbb{Z}_q^{m\times n}$ and $b, s \in \mathbb{Z}_q^n$ be drawn uniformly at random, and $e \in \mathbb{Z}_q^m$ drawn according to χ.

Distinguish between (*A*, *As* + *e*), and (*A*, *b*).

Search and Decision variants

LWE (decisional). Let $s \in \mathbb{Z}_q^n$ be drawn uniformly at random. Given access to either $O_\$$ or O_s, distinguish between the two.

LWE (search). Let $s \in \mathbb{Z}_q^n$ be drawn uniformly at random. Given access to either O_s, find *s*.

Proposition 1: the two problems are equivalent up to polynomial reductions (“hybrid” technique).

Proposition 2: given an efficient algorithm that solves SIS with parameters *n*, *m*, *q*, β, there is an efficient algorithm that solves LWE with the same parameters, assuming (roughly) $\alpha\beta \ll 1$.

LWE and BDD

LWE (bounded samples). Let $A \in \mathbb{Z}_q^{m\times n}$ and $b, s \in \mathbb{Z}_q^n$ be drawn uniformly at random, and $e \in \mathbb{Z}_q^m$ drawn according to χ.

Distinguish between (A, $As + e$), and (A, b).

Proposition 3: LWE reduces to BDD with $\gamma = q^{n/m}/\alpha$.

Consider the lattice $\mathscr{L} = A\mathbb{Z}_q^n$ generated by A.

The shortest vector is expected to have norm $\lambda_1(A) \sim \sqrt{(m)}q^{(m-n)/m}$.

The standard deviation of e is $\sqrt{m}\alpha q$.

(In particular we can expect the closest lattice point to $As+e$ *is* As.)

Better! Regev '05: Solving LWE (for uniformly random A) implies **quantumly** solving GapSVP in dimension n for **any** lattice!

→ "Worst-case to average-case" reduction. Note m irrelevant.

Classical reduction in dim $\sqrt{n}$, Peikert '09.

Flexibility of LWE

Many variants of LWE reduce to LWE:

- Binary-LWE: s is in $\{0,1\}^n$ (with limited samples).

- Learning with Rounding (LWR): the error is uniform in a small range instead of Gaussian. Amounts to deterministic rounding!

- …

Can be used for a host of applications:

- Secret-key encryption, PRF.

- PKE, key exchange.

- Identity-based encryption (see Michel's course), FHE.

- …

Secret-key encryption using LWE

Like LPN:

Pick a secret s uniformly in $\mathbb{Z}_q^n$.

Secret key: s.

Encrypt: to encrypt one bit b: give $(a, a \cdot s + b\lfloor q/2 \rfloor + e)$.

Decrypt: compute $a \cdot s$ to retrieve $b\lfloor q/2 \rfloor + e$, output b=1 iff closer to $\lfloor q/2 \rfloor$ than to 0.

IND-CPA security sketch: $(a, a \cdot s + e)$ is indistinguishable from uniform, hence so is $(a, a \cdot s + b\lfloor q/2 \rfloor + e)$.

A public sampler for LWE

To make previous scheme public-key, we'd like a public "sampler" for LWE. Should not require knowing the secret s.

Setup:
- Pick a secret s uniformly in $\mathbb{Z}_q^n$.
- Publish m LWE(q,n,χ) samples for large enough m (value TBD).

That is, publish (A,As+e) for $m\times n$ matrix A.

Now to get a fresh LWE sample:
- Pick x uniformly in $\{0,1\}^n$.
- Publish (${}^t xA$, ${}^t x(As+e)$).

With the right parameters, this yields a distribution statistically close to LWE(q,n,χ'), where if χ is Gaussian with variance σ^2, χ' is Gaussian with variance $m\sigma^2$.

Argument: Leftover Hash Lemma. Example: $m = 2n \log q$ suffices.
Remark: recognize the Ajtai hash function from earlier/subset sum.

Public-key encryption* using LWE

Regev '05: **Regev encryption.**
Idea: same as secret-key scheme, but with public sampler.

Pick a secret s uniformly in $\mathbb{Z}_q^n$, A uniformly in $\mathbb{Z}_q^{m\times n}$.

Public key: $(A, b = As + e)$.

Secret key: s.

Encrypt: to encrypt one bit k: draw x in $\{0,1\}^m$, output:

$$({}^t xA, {}^t xb + k\lfloor q/2 \rfloor).$$

Decrypt: upon receipt of ciphertext (c,d), output 0 if $d-c\cdot s$ is closer to 0 than to $\lfloor q/2 \rfloor$, 1 otherwise.

Proof argument. Step 1: public key is indistinguishable from uniform. Step 2: assuming uniform public key, ciphertexts are statistically close to uniform.

*malleability → not IND-CCA.

Practical (in)efficiency

Example parameters: q prime $\cong n^2$, $m = 2\,n \log q$, $\alpha = 1/(\sqrt{n}\log^2 n)$.
In practice, e.g. n$\cong$200.

Terrible efficiency:
- O(n^2) operations for encryption.
- O($n \log n$) ciphertext for 1 bit of plaintext!

Multi-bit Regev encryption

Idea: use multiple secrets.

Pick a secret **matrix** S uniformly in $\mathbb{Z}_q^{\ell \times n}$, A uniformly in $\mathbb{Z}_q^{m \times n}$.

Public key: $(A, B = AS + E)$.

Secret key: S.

Encrypt: to encrypt ℓ bits $k \in \{0,1\}^\ell$: draw x in $\{0,1\}^m$, output:

$$({}^t xA, {}^t xB + \lfloor q/2 \rfloor k).$$

Decrypt: upon receipt of ciphertext (C,D), output $k \in \{0,1\}^\ell$ such that $D\text{-}C \cdot S$ is closest to $\lfloor q/2 \rfloor k$.

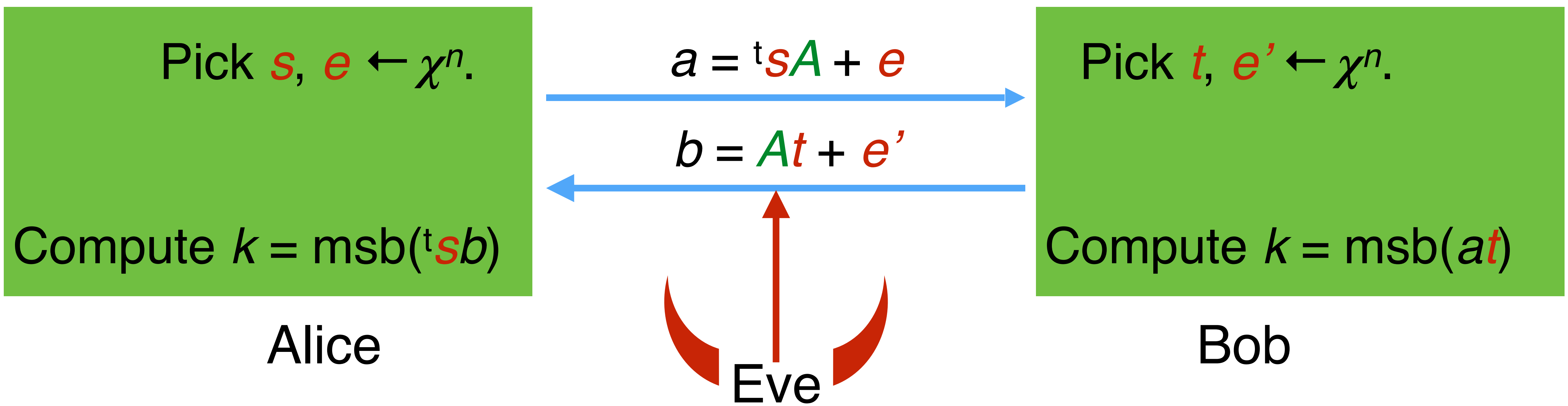
Proof argument: use multiple-secret LWE.

Ciphertext expansion $(n/\ell + 1) \log q$.

Other idea: encode multiple bits per element in $\mathbb{Z}_q$. (use high-order bits.)

Key exchange

Setup: pick public A uniformly in $\mathbb{Z}_q^{n\times n}$.



Here, msb = most significant bit.

Both parties get ${}^t s A t$ up to error terms. msb gets rid of error.

Equivalent of DDH: Eve wants to distinguish (A,a,b,k) from $(A,\$,\$,\$)$.

Proof argument: 1st hybrid $(A,\$,b,k)$. 2nd hybrid $(A,\$,\$,\$)$. Use LWE with secret-error switching on A, then $(A|a)$.

Practical aspects

Improving efficiency: compressing A

LWE (decisional). Let $s \in \mathbb{Z}_q^n$ be drawn uniformly at random.
Distinguish $(a, a \cdot s + e)$ from (a, b) for uniform a, b, and $e \leftarrow \chi$.

To get one "usable" b you need to publish the corresponding a, which is n times larger.

It'd be nice if the matrix A of a's was structured → compressible.

Simple idea: cyclic A. (See cyclic codes…)

Amounts to operating in ring $\mathbb{Z}_q[X]/(X^n - 1)$ → **Ring-LWE**.

Ring-LWE

Let R = $\mathbb{Z}_q[X]/P$ for some polynomial *P* (think irreducible).

Ring-LWE (decisional). Let $s \in R$ be drawn uniformly at random. Distinguish $(a, a \cdot s + e)$ from (a, b) for uniform a, $b \leftarrow R$, and $e \leftarrow \chi$.

The "usable" part *b* is now the same size as the uniform part *a*.

Example: **Regev encryption**
- ciphertext expansion O(1) instead of O(n).
- with proper choice of ring (e.g. arising from cyclotomic polynomials), $a \cdot s$ can be computed in $n \log n$, not n^2, using FFT.

Theoretical concern: reduces to hard *ideal* lattice problems. Believed to be as hard as general case, beside a few "trivial" properties (e.g. SVP = SIVP, collision on Ajtai hash function).

Concrete security

For factorization or Discrete Log, essentially one *family* of attacks.

For LWE and other lattice-based schemes, much more difficult:
- lattice reduction algorithms: LLL, BKZ.
- BKW-type algorithms (connection with LPN).
- ISD algorithms (connection with decoding random code).
- For low errors, such as Arora-Ge and Gröbner bases (connection with multivariate system solving).

→ ongoing NIST standardization process to fix concrete parameters.