Reasons to study post-quantum cryptography:

1. Protect confidentiality of today's computations against potential future threat
2. Standards take a long time to develop and deploy, so should start now
   - ↳ NIST has initiated a multi-year initiative to develop and standardize post-quantum key-exchange and signatures (currently in 2nd year of 6-year initiative)
   - ↳ Google recently piloted an experiment involving post-quantum key exchange in Chrome (using a "best of both worlds" approach where key derived from mix of classic key exchange and post-quantum key exchange)
3. New kinds of mathematical structures and assumptions — opportunity to build cryptography up from scratch again!

<u>Our focus</u>: lattice-based cryptography

<u>Definition</u>. An n-dimensional lattice $\mathcal{L}$ is a "discrete additive subspace" of $\mathbb{R}^n$:

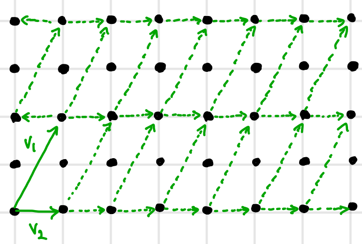
1. Discrete: every $x \in \mathbb{R}^n$ has a neighborhood in $\mathbb{R}^n$ where it is the <u>only</u> point
2. Additive subspace: $0^n \in \mathcal{L}$ and for all $x, y \in \mathcal{L}$, $-x \in \mathcal{L}$ and $x+y \in \mathcal{L}$

<u>Example</u>: the integer lattice $\mathbb{Z}^n$, the "q-ary" lattice $q\mathbb{Z}^n$ (i.e., the set of vectors where each entry is an integer multiple of q)

While most (non-trivial) lattices are infinite, they are finitely-generated by taking <u>integer</u> linear combinations of a finite collection of basis vectors $B = \{b_1, \dots, b_k\}$:

$$\mathcal{L} = \mathcal{L}(B) = B \cdot \mathbb{Z}^k = \left\{ \sum_{i \in [k]} \alpha_i b_i : \alpha_i \in \mathbb{Z} \text{ for all } i \in [k] \right\}$$

Example over $\mathbb{R}^2$:



<u>Computational problems</u>:

- Shortest vector problem (SVP): Given a basis B for a lattice $\mathcal{L} = \mathcal{L}(B)$, find a shortest <u>non-zero</u> vector $v \in \mathcal{L}$

  for simplicity, we will use the ℓ_2-norm

- Approximate SVP (SVP_γ): Given a basis B for a lattice $\mathcal{L} = \mathcal{L}(B)$, find a non-zero vector $v \in \mathcal{L}$ such that $\|v\| \le \gamma \cdot \lambda_1(\mathcal{L})$, where $\lambda_1(\mathcal{L})$ denotes the norm of the shortest non-zero vector in $\mathcal{L}$

  approx factor typically function of lattice dimension n

- Decisional approximate SVP ($\text{GapSVP}_{d,\gamma}$): Given a basis B for a lattice $\mathcal{L} = \mathcal{L}(B)$ where either $\lambda_1(\mathcal{L}) \le d$ or $\lambda_1(\mathcal{L}) \ge \gamma \cdot d$, decide which is the case

Many other lattice problems, but these should provide a flavor for what lattice problems look like

Approximation factor γ determines hardness of problem:

- $\gamma = O(1)$: NP-hard
- $\gamma = \tilde{O}(n)$: useful for cryptographic constructions
- $\gamma = 2^{n \log\log n / \log n}$: polynomial time

$\Big\}$ for lattice dimension n

Learning with Errors (LWE): The LWE problem is defined with respect to lattice parameters n, m, q, χ, where χ is an error distribution over $\mathbb{Z}_q$ (oftentimes, this is a discrete Gaussian distribution over $\mathbb{Z}_q$). The $\mathrm{LWE}_{n,m,q,\chi}$ assumption states that for a random choice $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $s \xleftarrow{R} \mathbb{Z}_q^n$, $e \leftarrow \chi^m$, the following two distributions are computationally indistinguishable:

$$(A, s^T A + e^T) \overset{c}{\approx} (A, r)$$

where $r \xleftarrow{R} \mathbb{Z}_q^m$.

LWE as a lattice problem: The search version of LWE essentially asks one to find s given $s^T A + e^T$. This can be viewed as solving the "bounded-distance decoding" (BDD) problem on the q-ary lattice

$$\mathcal{L}(A^T) = \{ s \in \mathbb{Z}_q^n : A^T s \} + q\mathbb{Z}^n$$

i.e., given a point that is close to a lattice element $s \in \mathcal{L}(A^T)$, find the point s

Symmetric encryption from LWE (for binary-valued messages) [Regev]

Setup(1^λ): Sample $s \xleftarrow{R} \mathbb{Z}_q^n$.

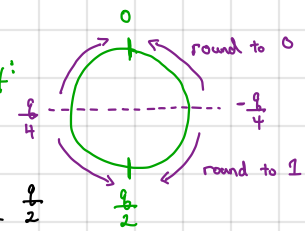
Encrypt(s, μ), $\mu \in \{0,1\}$: Sample $a \xleftarrow{R} \mathbb{Z}_q^n$ and $e \leftarrow \chi$. Output $(a, s^T a + e + \mu \cdot \lfloor \frac{q}{2} \rfloor)$.

Decrypt(s, ct): Output $\underbrace{\lfloor ct_2 - s^T ct_1 \rceil_2}_{\text{"rounding operation"}}$

$$\lfloor x \rceil_2 = \begin{cases} 0 & \text{if } -\frac{q}{4} < x < \frac{q}{4} \\ 1 & \text{otherwise} \end{cases}$$

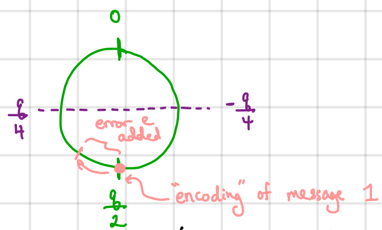
take $x \in \mathbb{Z}_q$ to be representative between $-\frac{q}{2}$ and $\frac{q}{2}$

Visually:



Correctness: $ct_2 - s^T ct_1 = s^T a + e + \mu \cdot \lfloor \frac{q}{2} \rfloor - s^T a$
$= \mu \cdot \lfloor \frac{q}{2} \rfloor + e$

if $|e| < \frac{q}{4}$, then decryption recovers the correct bit



(message encrypted in "most significant bits" of the ciphertext)
↳ will see variant in HW5

Security: By the $\mathrm{LWE}_{n,q,\chi}$ assumption, $(a, s^T a + e) \overset{c}{\approx} (a, r)$ where $r \xleftarrow{R} \mathbb{Z}_q$. Thus,

$$(a, s^T a + e + \mu \cdot \lfloor \tfrac{q}{2} \rfloor) \overset{c}{\approx} (a, r + \mu \cdot \lfloor \tfrac{q}{2} \rfloor)$$

$r \xleftarrow{R} \mathbb{Z}_q$: one-time pad encryption of the message μ

Observe: this encryption scheme is additively homomorphic (over $\mathbb{Z}_2$):

$$\begin{matrix} (a_1, s^T a_1 + e_1 + \mu_1 \cdot \lfloor \frac{q}{2} \rfloor) \\ (a_2, s^T a_2 + e_2 + \mu_2 \cdot \lfloor \frac{q}{2} \rfloor) \end{matrix} \Rightarrow \left(a_1 + a_2,\ s^T(a_1 + a_2) + (e_1 + e_2) + (\mu_1 + \mu_2) \cdot \lfloor \tfrac{q}{2} \rfloor\right)$$

decryption then computes

$$(\mu_1 + \mu_2) \cdot \lfloor \tfrac{q}{2} \rfloor + e_1 + e_2$$

which when rounded yields $\mu_1 + \mu_2 \pmod 2$ provided that $|e_1 + e_2 + 1| < \frac{q}{4}$

Idea: We will include encryptions of 0 in the public key and refresh ciphertexts by taking a subset sum of encryptions of 0:

Regev's public-key encryption scheme

Setup: $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ output $pk = (A, b^T)$

$s \xleftarrow{R} \mathbb{Z}_q^n$ $b^T \leftarrow s^T A + e^T$ $sk = s$

$e \leftarrow \chi^m$

$\uparrow$ can be viewed as m encryptions of 0 under the symmetric scheme with secret key s

Encrypt (pk, μ): sample $r \xleftarrow{R} \{0,1\}^m$

output $(Ar,\ b^T r + \mu \cdot \lfloor \frac{q}{2} \rceil)$

Decrypt (sk, ct): output $\lfloor ct_2 - s^T ct_1 \rceil_2$

Correctness: $ct_2 - s^T ct_1 = b^T r + \mu \cdot \lfloor \frac{q}{2} \rceil - s^T A r = s^T A r + e^T r + \mu \cdot \lfloor \frac{q}{2} \rceil - s^T A r$

$= \mu \cdot \lfloor \frac{q}{2} \rceil + e^T r$

if $|e^T r| < \frac{q}{4}$, then decryption succeeds (since e is small and r is binary, $e^T r$ is not large: $|e^T r| < m\|e\|\|r\| = m\|e\|$)

Security (Sketch): Under LWE assumption public key

$(A, s^T A + e^T) \approx (A, u)$ where $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $u \xleftarrow{R} \mathbb{Z}_q^m$

By the "leftover hash lemma," if we sample $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $u \xleftarrow{R} \mathbb{Z}_q^n$, $r \xleftarrow{R} \{0,1\}^m$ where $m > 2n \log q$

$(Ar, u^T r) \approx (v, w)$ where $v \xleftarrow{R} \mathbb{Z}_q^n$ and $w \xleftarrow{R} \mathbb{Z}_q$

$\Rightarrow$ $b^T r$ in ciphertext functions as a one-time pad