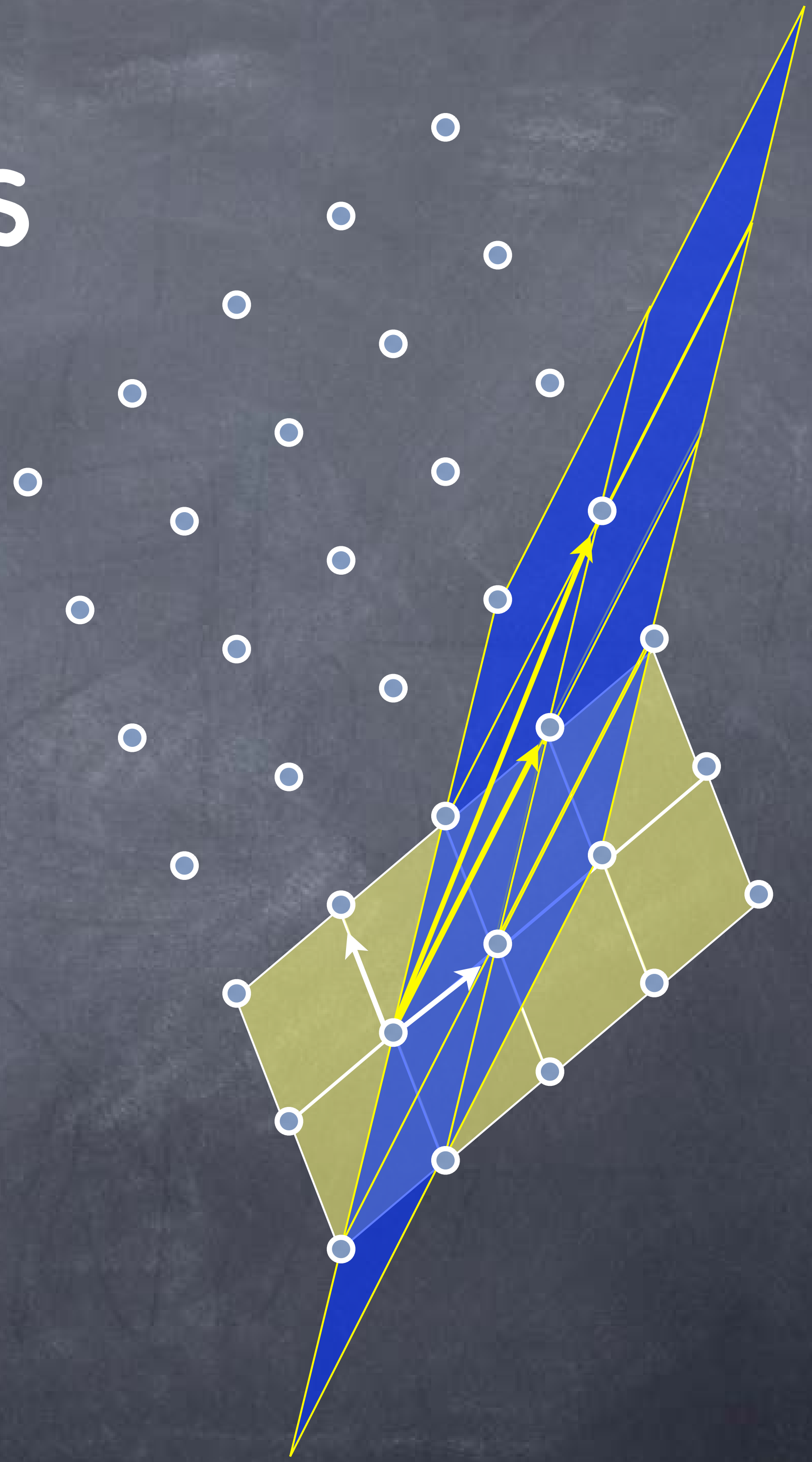
Lattice Cryptography

Lecture 19

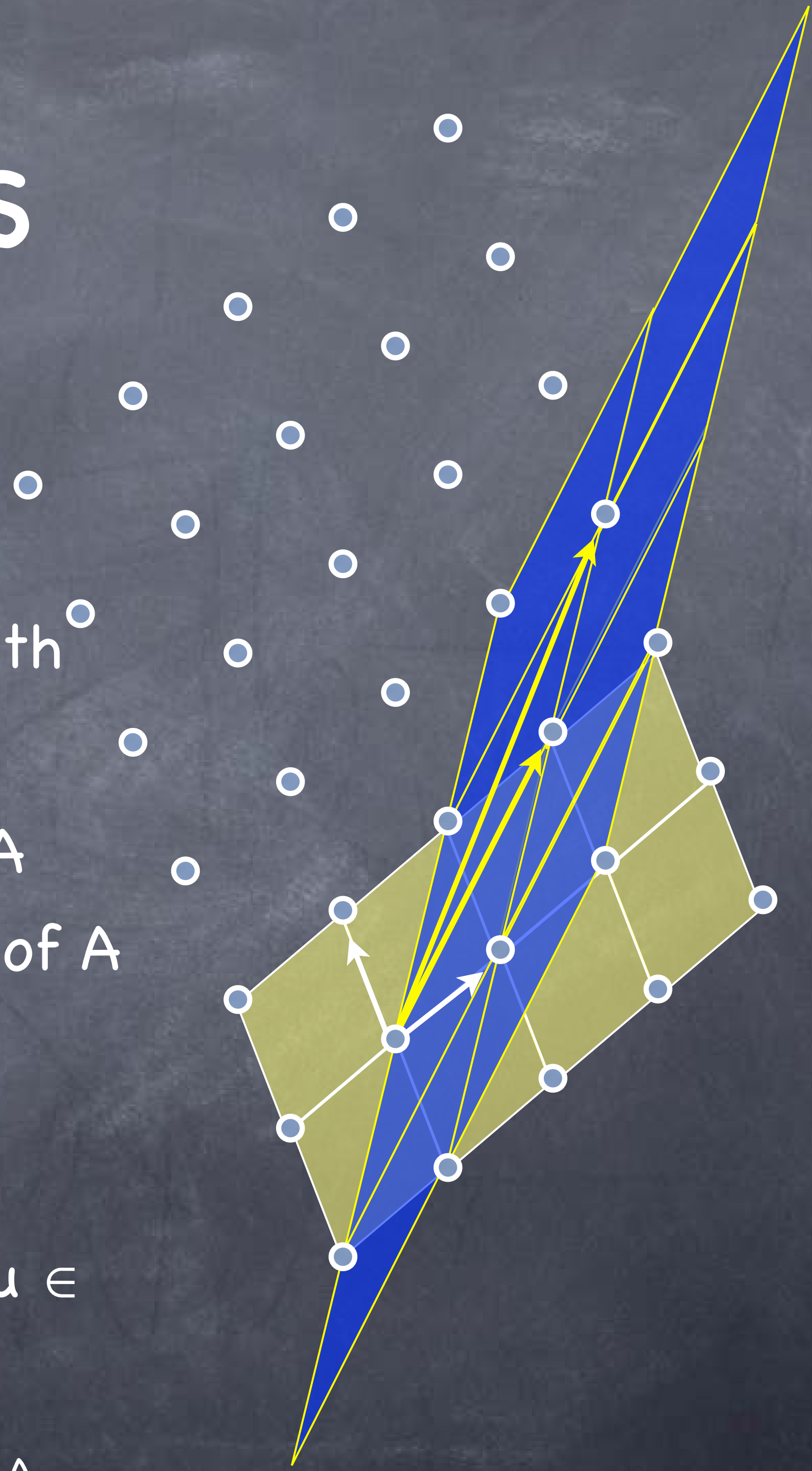
Lattices

- A infinite set of points in $\mathbb{R}^n$ obtained by tiling with a "basis"
  - Formally, $\{ \sum_i x_i \underline{\mathbf{b_i}} \mid x_i \text{ integers} \}$
- Basis is not unique
- Several problems related to high-dimensional lattices are believed to be hard, with cryptographic applications
  - Hardness assumptions are "milder" (worst-case hardness)
  - Believed to hold even against quantum computation: "Post-Quantum Cryptography"



Lattices

- Given a basis $\{\underline{\mathbf{b}_1},\ldots,\underline{\mathbf{b}_m}\}$ in $\mathbb{R}^n$, lattice has points $\{ \sum_i x_i \underline{\mathbf{b}_i} \mid x_i \text{ integers} \}$
- An interesting case: lattices in $\mathbb{Z}^n$
  - Two n-dim lattices in $\mathbb{Z}^n$ associated with an mxn matrix A over $\mathbb{Z}_q$
    - L_A : Vectors "spanned" by rows of A
    - $L_A^\perp$: Vectors "orthogonal" to rows of A
    - Here, L_A, $L_A^\perp$ in $\mathbb{Z}^n$, but above operations mod q (i.e., over $\mathbb{Z}_q$)
- Dual lattice L^*: $\{ \underline{\mathbf{v}} \mid \langle \underline{\mathbf{v}},\underline{\mathbf{u}} \rangle$ is an integer, $u \in L \}$
  - e.g. $(L_A)^* = 1/q\ L_A^\perp$ and $(L_A^\perp)^* = 1/q\ L_A$



Lattices in Cryptography

- Several problems related to lattices (lattice given as a basis) are believed to be computationally hard in high dimensions
- Closest Vector Problem (CVP): Given a point in $\mathbb{R}^n$, find the point closest to it in the lattice
- Shortest Vector Problem (SVP): Find the shortest non-zero vector in the lattice
  - SVP_γ: find one within a factor γ of the shortest
  - GapSVP_γ: decide if the length of the shortest vector is ≤ 1 or $\geq \gamma$ (promised to be one of the two)
  - uniqueSVP_γ: SVP, when guaranteed that the next (non-parallel) shortest vector is longer by a factor γ or more
- Shortest Independent Vector Problem (SIVP): Find n independent vectors minimizing the longest of them

Lattices in Cryptography

- Worst-case hardness of lattice problems (e.g. GapSVP)

NP-hard		in co-NP	in P

γ: 1 $\quad 2^{(\log n)^{\wedge}(1-\varepsilon)}$ $\quad \sqrt{n}$ $\quad n$ (crypto regime) $\quad 2^n$

- Assumptions about worst-case hardness (e.g. P≠NP) are qualitatively simpler than that of average-case hardness
  - Crypto requires average-case hardness
  - For many lattice problems average-case hardness implied by worst-case hardness of related problems

Average-Case/Worst-Case Connection

- Worst-case hardness: Hard to solve every instance of the problem (holds even if most instances are easy)
- Crypto typically needs average case hardness assumption: Random instance of a problem is hard to solve (broken if an algorithm can solve many instances)
- Worst-case connection: Show that solving random instances of Problem 1 is as hard as solving another (hard) problem Problem 2 in the worst case
- Connection shows that if a few instances (of the second problem) are hard, most instances (of the first problem) are
- For many lattice problems average-case hardness assumptions are implied by worst-case hardness of related problems (but at regimes not known to be NP-hard)

Hash Functions and OWF

- CRHF: $f(\underline{\mathbf{x}}) = A^T\underline{\mathbf{x}} \pmod q$
  - $\underline{\mathbf{x}}$ required to be a "short" vector (i.e., each co-ordinate in the range [0,d-1] for some small d)
  - A^T is an n x m matrix: maps m log d bits to n log q bits (for compression we require $m > n \log_d q$)
  - Collision yields a short vector (co-ordinates in [-(d-1),d-1]) $\underline{\mathbf{z}}$ s.t $A^T\underline{\mathbf{z}} = 0$: i.e., a short vector in the lattice $L_A^{\perp}$
  - Simple to compute: if d small (say, d=2, i.e., $\underline{\mathbf{x}}$ binary), $f(\underline{\mathbf{x}})$ can be computed using O(n m) additions mod q
- If sufficiently compressing (say by half), a CRHF is also a OWF

Short Integer Solution Problem

Has a worst-case connection to lattice problems

Succinct Keys

- The hash function is described by an n x m matrix over $\mathbb{Z}_q$, where n is the security parameter and m > n
  - Large key and correspondingly large number of operations
- Using "ideal lattices" which have more structure:
  - A random basis for such a lattice can be represented using just m elements of $\mathbb{Z}_q$ (instead of mn)
  - Matrix multiplication can be carried out faster (using FFT) with $\tilde{O}(m)$ operations over $\mathbb{Z}_q$ (instead of O(mn))
- Security depends on worst-case hardness of same problems as before, but when restricted to ideal lattices

Public-Key Encryption

- NTRU approach: Private key is a "good" basis, and the public key is a "bad basis"
  - Worst basis (one that can be efficiently computed from any basis): Hermite Normal Form (HNF) basis
- To encrypt a message, encode it (randomized) as a short "noise vector" u. Output c = v+u for a lattice point v that is chosen using the public basis
  - To decrypt, use the good basis to find v as the closest lattice vector to c, and recover u=c-v
- Use lattices with succinct basis (defined over the ring of degree N TRUncated polynomials)
- Conjectured to be CPA secure for appropriate lattices. No security reduction known to simple lattice problems

Learning With Errors

- LWE (computational version): given noisy inner-products of random vectors with a hidden vector, find the hidden vector
  - Given $<\underline{\mathbf{a}}_1,\underline{\mathbf{s}}>+\underline{\mathbf{e}}_1$, ..., $<\underline{\mathbf{a}}_m,\underline{\mathbf{s}}>+\underline{\mathbf{e}}_m$ and $\underline{\mathbf{a}}_1,\ldots,\underline{\mathbf{a}}_m$ find $\underline{\mathbf{s}}$.
    $\underline{\mathbf{a}}_i$ uniform, $\underline{\mathbf{e}}_i$ Gaussian noise (rounded, in $\mathbb{Z}_q$)
- If m fixed a priori: Given ($A\underline{\mathbf{s}}+\underline{\mathbf{e}}$, A) find $\underline{\mathbf{s}}$ where $A \in \mathbb{Z}_q^{m\times n}$
- Decision version: distinguish between such an input and a random input
- Assumed to be hard (note: average-case hardness). Has been connected with worst-case hardness of GapSVP
  - Turns out to be a very useful assumption

Learning With Errors

- (Decision) LWE is a fairly strong assumption that subsumes some other (more traditional) lattice assumptions
- Hardness of (Decision) LWE $\Rightarrow$ Hardness of Short Integer Solution
- Given algorithm for SIS, an algorithm for D-LWE: i.e, given $(A,\underline{\mathbf{b}})$, to check if $\underline{\mathbf{b}}=A\underline{\mathbf{s}}+\underline{\mathbf{e}}$ for a short $\underline{\mathbf{e}}$:
  - Find a short solution $\underline{\mathbf{x}}$ for $A^T\underline{\mathbf{x}} = 0$. Check if $\langle\underline{\mathbf{x}},\underline{\mathbf{b}}\rangle$ is short
  - If $\underline{\mathbf{b}}=A\underline{\mathbf{s}}+\underline{\mathbf{e}}$ then, $\langle\underline{\mathbf{x}},\underline{\mathbf{b}}\rangle=\langle\underline{\mathbf{x}},\underline{\mathbf{e}}\rangle$, which is short. If $\underline{\mathbf{b}}$ random, then $\langle\underline{\mathbf{x}},\underline{\mathbf{b}}\rangle$ random (for non-zero $\underline{\mathbf{x}}$), and unlikely to be short.

Learning With Errors

- A simple Worst-case/Average-case connection of (Decision) LWE
- Worst-$\underline{\mathbf{s}}$ hardness ⇒ Average-$\underline{\mathbf{s}}$ hardness
  - Note: A is still random
  - Given arbitrary instance $(A,\underline{\mathbf{b}})$, define $\underline{\mathbf{b}}^* = \underline{\mathbf{b}} + A\underline{\mathbf{r}}$ for a random vector $\underline{\mathbf{r}}$. If $\underline{\mathbf{b}}=A\underline{\mathbf{s}}+\underline{\mathbf{e}}$, then $\underline{\mathbf{b}}^*=A\underline{\mathbf{s}}^*+\underline{\mathbf{e}}$, for random $\underline{\mathbf{s}}^*=\underline{\mathbf{s}}+\underline{\mathbf{r}}$. If $\underline{\mathbf{b}}$ random, $\underline{\mathbf{b}}^*$ random
  - So, run algorithm for average $\underline{\mathbf{s}}$ on $(A,\underline{\mathbf{b}}^*)$ and output its decision

Public-Key Encryption

- An LWE based approach:
  - Public-key is (A,P) where P=AS+E, for random matrices (of appropriate dimensions) A and S, and a noise matrix E over $\mathbb{Z}_q$
  - To encrypt an n bit message, first map it to a vector $\underline{\mathbf{v}}$ in (a sparse sub-lattice of) $\mathbb{Z}_q^n$; pick a random vector $\underline{\mathbf{a}}$ with small coordinates; ciphertext is ($\underline{\mathbf{u}}$,$\underline{\mathbf{c}}$) where $\underline{\mathbf{u}} = A^T\underline{\mathbf{a}}$ and $\underline{\mathbf{c}} = P^T\underline{\mathbf{a}} + \underline{\mathbf{v}}$
  - Dec(($\underline{\mathbf{u}}$,$\underline{\mathbf{c}}$),S): recover $\underline{\mathbf{v}}$ by "rounding" $\underline{\mathbf{c}} - S^T\underline{\mathbf{u}} = \underline{\mathbf{v}} + E^T\underline{\mathbf{a}}$
    - Allows a small error probability; can be made negligible by first encoding the message using an error correcting code
  - CPA security: By (Decision) LWE assumption, the public-key is indistinguishable from random; and, encryption under random (A,P) <u>loses essentially all information</u> about the message
    - If P uniform, (P,$P^T\underline{\mathbf{a}}$) is statistically close to uniform

Next time

Today

- Lattice based cryptography
  - Candidate for post-quantum cryptography
  - Security typically based on worst-case hardness of problems
  - Several problems: SVP and variants, LWE
  - Applications: Hash functions, PKE, ...
- Next: Fully Homomorphic Encryption