Lesson 11 Lecture Notes

CS555

1 Recap

- **Trapdoor Permutations**
  - One-way function with additional properties.
  - Has some trapdoor which, when known, makes the function easy to invert.
- RSA: based on DLP.

1.1 Goldwasser–Micali Encryption

- Based on quadratic residues.
- **Gen:** Choose two large primes p, q, set $N = pq$. Public key $pk = (N, y)$, secret key $sk = (p, q)$.
- **Enc:** For message bit b:
  1. Generate random $r \in \mathbb{Z}_N^*$.
  2. If $b = 0$, $c = r^2 \pmod N$ (quadratic residue).
  3. If $b = 1$, $c = r^2 y \pmod N$ (non-quadratic residue).
- **Dec:** Using (p, q), check if $c \in QR_N$.
  - note, if $c \in QR_N \to c \in QR_p, c \in QR_q$
- **IND-security** follows from the quadratic residuosity assumption: given N, no PPT can distinguish between random quadratic residues and non-quadratic residues with Jac_{+1}
- GM is homomorphic wrt $\oplus$:
  - Given GM-ciphertexts of bits b and b', one can compute a ciphertext of $b + b' \pmod 2$.
  - Computation on ciphertexts yields the same result as computation on plaintexts.
  - To compute $b \oplus b'$ from ciphertexts c and c', perform $c \cdot c'$.
  - That is:
$$\text{Enc}(pk, b) \cdot \text{Enc}(pk, b') = \text{Enc}(pk, b \oplus b') = \text{Enc}(pk, b + b' \bmod 2).$$

2 Post-Quantum Security and Lattice-Based Crypto

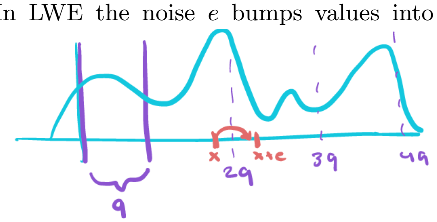
2.1 Motivation

- Best known algorithms for lattices run in 2^n time, while factoring/DLP can be solved in about $2^{\sqrt[3]{n}}$ (sub-exponential)
- so far, its quantum resistant
- Worst-case hardness strongly connected to average-case hardness.
- Simple and efficient constructions.
- enabler of other capabilities
- Fully Homomorphic Encryption (FHE)
- 1994: Shor gives quantum construction for factoring and DLP $\to$ capacity to actualize this is not yet available, but it is coming
- **Post Quantum Cryptography:** schemes that *should* be quantum-resistant

2.2 Learning With Errors (LWE)

- Setup: matrix A, secret s.
- Goal: recover s given noisy linear equations.
- Attempts:
  1. Find s given As (easy via Gaussian elimination).
  2. Find s given $As \pmod q$ (harder but still solvable).
  3. Find s given $As + e$ (solvable by linear regression).
  4. **Find s given $(As + e) \pmod q$ (LWE problem).**

- In LWE the noise e bumps values into new distribution regions; modulus condenses distribution into $[0, q-1]$.



- LWE Concept:
  - decoding random linear codes (error correcting code with error)
  - learning noisy linear functions
  - worst-case hard lattice problems

2.3 Attack 1: Linearization

- Goal: Given $(A, As + e)$, recover s.
- **Try 1:** Each linear equation is an exact polynomial equation
  - Let $b = \langle a, s \rangle + e = \sum_{i=1}^{n} a_i s_i + e$ (*Here the error bound is $B = 1$ giving $e \in \{-1, 0, 1\}$.
  - Have $0 = b - \sum a_i s_i - e \in \{-1, 0, 1\}$).
  - So $(b - \sum a_i s_i - 1)(b - \sum a_i s_i)(b - \sum a_i s_i + 1) = 0$.
  - One of these factors is zero, so the product is zero.
  - This yields a degree-2 polynomial equation.
- Even solving such degree-2 equations is NP-hard.
- **Try 2:** Use linearization to simplify.
  - Expand the polynomial and replace monomials with indexed variables.
  - For example, replace $s_i s_j s_k$ with x_{ijk}.
  - This creates a system of noiseless linear equations in new variables.
  - Fewer equations $\to$ more candidate solutions; more equations $\to$ fewer candidates.
  - When number of equations $\approx$ number of variables, solve via Gaussian elimination in $O(n^3)$.

2.4 Generalized Linearization Attack

- Breakable when number of equations $m >> n^{2B+1}$.
- Set $B = n^{\Omega(1)}$.
- To defend: ensure linearization doesn't yield enough equations to solve.

2.5 Lattice Concepts

- Lattice: discrete additive subgroup of $\mathbb{R}^n$.
- A is points on lattice
- e does not have to be on the lattice

2.6 Lattice Reduction Attack

- Use the LLL algorithm to attack LWE instances.
- Assume:
$$\frac{q}{B} = 2^{n^{\varepsilon}} \quad \text{for some constant } \varepsilon > 0.$$
- Then LLL solves LWE in time:
$$2^{\tilde{O}(n^{1-\varepsilon})} \cdot \mathrm{poly}(n, \log q).$$
- Runtime is polynomial in n and $\log q$ when:
$$\frac{q}{B} = 2^{\Omega(n)}.$$
- This gives intuition for why noise bounds B can't be too small — small B makes the attack easier.

2.7 Safe Parameters

- n: security parameter.
- m: polynomial in n.
- B: small polynomial in n (say $\sqrt{n}$.
- q: polynomial in n, larger than B, possibly sub-exponential (say $2^{0.99n}$.
- No known quantum algorithms to break well-constructed LWE.

2.8 Decisional LWE

- Can you distinguish between $(A, As + e)$ [Correlated items] and (A, b) [completely independent].
- Same hardness as LWE.
- Independence: for arbitrary s, cannot see dependence on A vs. random.

2.9 Info-Computation Gap

- If m (columns in A) is much smaller than the number of variables n, it becomes information-theoretically hard to recover s.
- However, even when s is uniquely determined by $(A, As + e)$, it may still be computationally hard to recover.
- There exists a range for m where this hardness holds:

$$\frac{n}{1 - \frac{\log(2B+1)}{\log q}} \leq m \leq \frac{n}{2^{\log\left(\frac{q}{2B+1}\right)}}.$$

- Within this range:
  - s is uniquely determined by the noisy measurements.
  - But recovering s remains computationally infeasible.

3 Applications of LWE

3.1 OWF and PRG

$$g_A(s, e) = As + e$$

- One-way function (by LWE).
- PRG (decisional LWE).
- Can be trapdoor.

3.2 Secret-Key Encryption

- **Gen:** $sk =$ vector $s \in \mathbb{Z}_q^n$.
- **Enc:** For message $m \in \{0, 1\}$:
  1. Sample $a \in \mathbb{Z}_q^n$, small noise $e \in \mathbb{Z}$.
  2. Compute $c = (a, b = \langle a, s\rangle + e + m \cdot \lfloor q/2 \rfloor)$.
- **Dec:** Output $\mathrm{round}_{q/2}(b - \langle a, s\rangle \pmod q)$.
- Correctness holds as long as $|e| < q/4$.