Post Quantum Cryptography

- There are certain computational problems that we suspect to be non polynomial-time for classical computers but polynomial time on a quantum computer (i.e. factoring)
  - For some computational problems, they give only polynomial speedups at best
- In this class, we will focus on honest parties using classical computers, but adversaries using quantum computers
- It very much looks like classical cryptography but we have to:
  - Base our cryptography on computational problems that seem hard for both classical anmd quantum computers
  - Set the parameters of our schemes (i.e. secret key lengths) to defeat quantum attacks
- Even though current quantum computers are far from being made:
  - People can use store-now-decrypt-later attacks
  - The U.S. government will force suppliers to use post-quantum by 2027, so all major tech companies will support them

Grover's Search

We look at the following problem called **unstructured search**:

- Given a function $f:\{0,1\}^n \to \{0,1\}$, find a value x such that $f(x)=1$ if one exist
- With a classical computer, this requires a $\Sigma(2^n)$ brute force approach

Grover gives a quantum circuit with size $\text{poly}(n)|f|\cdot 2^{n/2}$ where $|f|$ denotes the size of f as a Boolean circuit

- If f is efficient, this is roughly size $2^{n/2}$
- For $n=128$, a classical computer takes 2^{128} invocations of f while a quantum circuit uses 2^64 invocations

Decryption under Known-Plaintext

Suppose we have a symmetric key encryption scheme that takes an n bit key k and a $10n$ bit message m, with a $10n$ bit ciphertext

- Suppose we have a ciphertext and we know the key k is unique
- We define a function that is 1 if x is a valid key to produce the cipher text and 0 otherwise

Grover Search can find the key in time $2^{n/2}$, giving a square root speedup

- Therefore, if we are aiming for 128 bit security, we need to use at least 256 bit encryption keys

Shor's Algorithm

We look at the problem **period finding**:

- Suppose we have integers d and N with a fucntion $f:\mathbb{Z}_N^d \to \mathbb{Z}_N^d$
- We want to find a non-zero value $\Delta \in \mathbb{Z}_N^d$ such that for all x, we have $f(x)=f(x+\Delta)$

Clasically, this requires super-polynomially many queries to f and apparently exponential when d is large

- A quantum circuit of size \text{poly(\log N)} can solve this problem with good probability, which is an exponential speedup!

Discrete Log Problem

The discrete log problem is a classic one from cryptography given as:

- Given a group G of prime order q with generator g and a problem instance $h \in G$ for a random $h \in G$
- Find the unique value $z \in \mathbb{Z}_q$ such that $h = g^z$

This is crucial for factoring numbers in RSA via Euler's Totient Theorem, and Shor's Algorithm gives a way to use period finding to solve it