Week 7: Shor's Algorithm for RSA

SHOR'S ALGORITHM

Shor's algorithm is a **quantum algorithm** that can factor large integers **exponentially faster** than the best-known classical algorithms.

STEPS

Step 1: Pick a Random Integer

Pick a random number a, such that:

- 1 < a < N
- gcd (a,N) = 1 (if gcd(a,N)≠1, we already found a factor)

Step 2: Period Finding (Quantum Step)

Use a quantum computer to find the **period** r of the function: f(x) = a^x mod N

That means: a^r ≡ 1 mod N

This step is what Shor's algorithm does efficiently using **Quantum Fourier Transform (QFT)**.

Step 3: Check r

If r is **even** and a^{r/2} ≠ -1 mod N, continue.

Step 4: Compute GCDs (Back to Classical)

Now compute:

gcd(a^{r/2} - 1, N) and gcd(a^{r/2} + 1, N)

At least one of them should give a **nontrivial factor** of N — which is either p or q.

Why This Breaks RSA

If you can factor N, you can compute ϕ(N), then compute the private key d, and **break the encryption**.

Shor's algorithm reduces the time to factor N from exponential to **polynomial** — which is why **RSA is insecure** against quantum computers.