Lecture 14 CPSC 447/547 - Intro to QC.

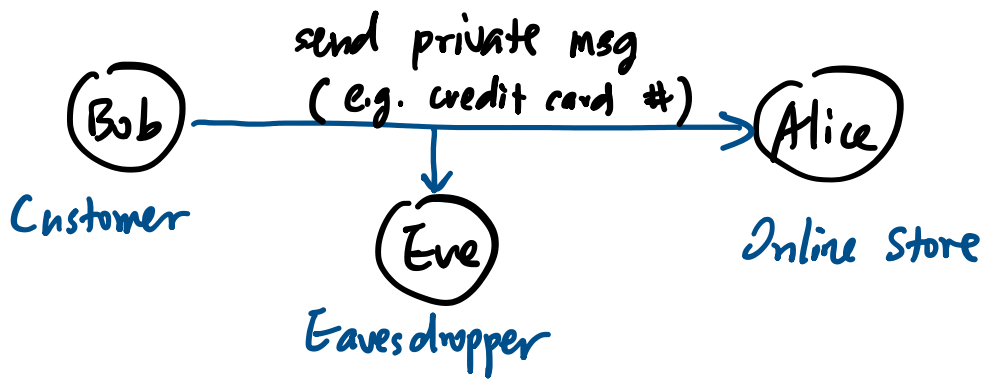
RSA and Shor's factoring algorithm

Outline

- Brief intro to RSA scheme
- QFT and Order finding.

RSA (Rivest-Shamir-Adleman)

A Public-key encryption scheme.



- Bob wants the msg to be decryptable by Alice only.
- Alice picks two large prime numbers, p and q (private)

  publish $N = p \cdot q$ (N is public)

  also publish e (e is public)

  msg^e and N

$$\begin{tikzcd}
\text{Plaintext} \quad m \arrow[r, bend left, "\text{Encrypt}"] & \text{Ciphertext} \quad c \arrow[l, bend left, "\text{Decrypt}"]
\end{tikzcd}$$

$c \equiv m^e \bmod N$

$m \equiv c^d \bmod N$

(basic number theory)

- d is computed efficiently, knowing p and q.
  but challenging to compute if not.
- If one can factor $N = p \cdot q$, RSA can be broken.
- RSA-512: $n = \log N = 512$ bits, RSA-1024: 1024 bits
- Shor's algorithm can factor N in poly(n) time.
- Naive classical algorithm (check divisors) $\approx \sqrt{2^n}$ time.

Prime Factorization.

Given a large number N, assume $N = p \cdot q$.
for some prime number p and q.
Find p and q.

Shor's observation:

Use period-finding (inspired by Simon's algorithm)
and quantum fourier transform (over $\mathbb{Z}_N$)

Why? A known fact from '70: "order-finding"

Proposition ("Square-root-of-1")

If we can find a (non-trivial) r ← $r \not\equiv \pm 1$

such that $r^2 \equiv 1 \bmod N$, then we can factor N.

Proof. $r^2 - 1 \equiv 0 \bmod N$, $N = p \cdot q$

$\Rightarrow p, q \mid (r+1)(r-1)$ ← divides

but $N \nmid (r+1)$ or $N \nmid (r-1)$ ← $r \not\equiv \pm 1 \bmod N$

$\Rightarrow p \mid (r+1)$ and $q \mid (r-1)$ or $p \mid (r-1)$ and $q \mid (r+1)$

$\Rightarrow \gcd(N, r+1)$ and $\gcd(N, r-1)$ are nontrivial factors! ∎.

Goal: Find such r! (order-finding)

- Pick a number $x \in \mathbb{Z}_N$ uniformly random.

  (if $\gcd(x, N) = k \neq 1$, we found a factor: k ☺)

  (so assume $\gcd(x, N) = 1$.)

Proposition: With "good" probability, this random x:

$x^s \equiv 1 \bmod N$ and s is even.

and $x^{s/2} \not\equiv 1 \bmod N$

This means x^s is such non-trivial "square root of 1"!

A quantum algorithm for finding a valid s?

Observation:

Periodic function $f(s) = x^s \bmod N$: "modular exponentiation".

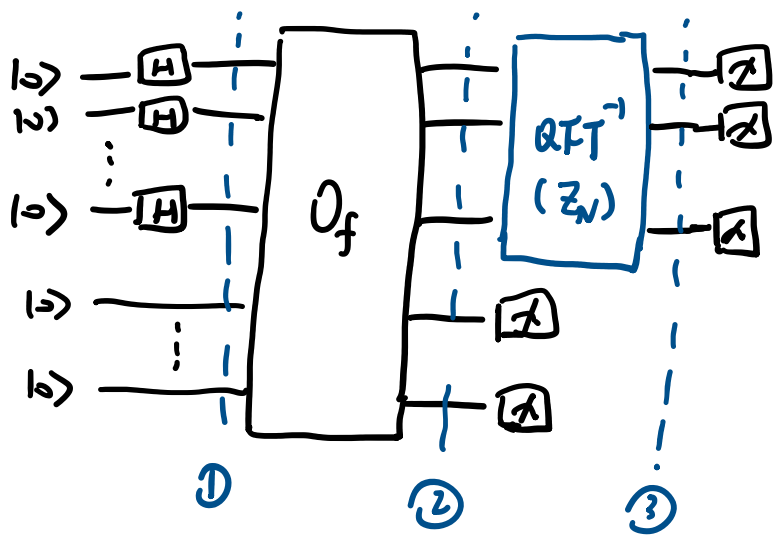
$s^* =$ order of x in $\mathbb{Z}_N$: least s s.t. $x^s \equiv 1 \bmod N$.

$$x \bmod N,\ x^2 \bmod N,\ x^3 \bmod N,\ \cdots,\ x^{r^*} \equiv 1 \bmod N.$$

$$\equiv \qquad \equiv \qquad \equiv$$

$$x^{r^*+1} \bmod N,\ x^{r^*+2} \bmod N,\ x^{r^*+3} \bmod N,\ \cdots$$

Hint: use Simon's algorithm ($\mathbb{Z}_N$ version) to find period s in f.



① $\frac{1}{\sqrt{N}} \sum_s |s\rangle |0\rangle$ "uniform superposition"

② $\frac{1}{\sqrt{N}} \sum_s |s\rangle |f(s)\rangle$

If we measure the output register and get a random $|f(s)\rangle = |0\rangle$

$|f(s)\rangle = |c\rangle$.

State: $\frac{1}{\sqrt{N}} \sum_{s:\, f(s)=c} |s\rangle|c\rangle$

E.g. $\frac{1}{\sqrt{N}} \left(|s\rangle |x^s \bmod N\rangle + |s+r^*\rangle |x^s \bmod N\rangle + |s+2r^*\rangle |x^s \bmod N\rangle \cdots \right)$

③ Then using QFT^{-1} (over $\mathbb{Z}_N$)

Just like in Simon's, this gives info on period r^*.

<u>Remaining</u>: How to implement O_f for modular exponentiation?

[Reading: N&C Ch 5.3].