Ph/CS 219A
Quantum Computation

Lecture 15. Factoring, Public Key Crypto, Phase Estimation

Last time we saw that the Quantum Fourier Transform (QFT) can be executed efficiently on a quantum computer, and that the QFT can be used to find the period of a black box function using a constant number of queries to the box.

This Period Finding algorithm is the key ingredient in Shor's algorithm for factoring integers, as we'll see today. We will also discuss the impact of Shor's algorithm in the security of public key cryptography protocols that are in wide use today.

Then we'll put Period Finding in a broader context, noting that the same method can be used to estimate the eigenvalues of a unitary matrix ("phase estimation").

An updated version of the Chapter 6 Lecture Notes has been posted on the course website.
Problem Set 4 has been posted, due December 4.

Order Finding

The quantum part of Shor's factoring algorithm is the subroutine for finding the period of an efficiently computable function, which we have already discussed. Relating period finding to factoring is an exercise in number theory, and there is nothing intrinsically quantum about it. But the connection is interesting and of societal importance, so let's discuss it.

$$\mathbb{Z}_N^* = \text{multiplicative group mod } N = \{b \in \mathbb{Z} \,|\, 1 \le b < N, \quad \text{such that} \quad GCD(N,b)=1\}$$

Inverse exists: Consider $\{ab \quad (\text{mod}\, N), \quad a \text{ fixed}, \quad b \in \mathbb{Z}_N^*\}$

These elements are distinct.

$$ab = ab' \quad (\text{mod } N) \Rightarrow ab - ab' = a(b-b') = N \times \text{integer} \Rightarrow (b-b') \quad \text{divides} \quad N \Rightarrow b \equiv b' \ (\text{mod}\, N).$$

Therefore, *ab*=1 (mod *N*) for some (unique) *b*. $\left|\mathbb{Z}_N^*\right| = \varphi(N)$ (Euler phi function).

$a \in \mathbb{Z}_N^*$ has order $\text{Ord}(a,N) = \min r$ such that $a^r = 1.$

$\text{Ord}(a,N) =$ period of function $f_{a,N}(x) = a^x \text{ mod } N.$

Order of *a* is period of this *modular exponential* function.

Furthermore, modular exponential is efficiently computable.

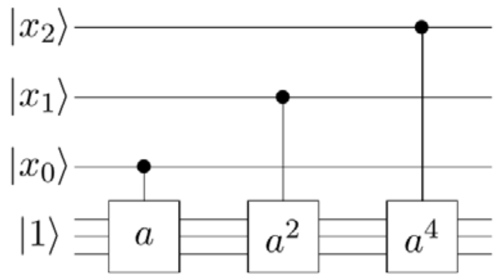
From Order Finding to Factoring

Modular exponential is efficiently computable by *repeated squaring*.

$$x = x_{m-1}\cdot 2^{m-1} + x_{m-2}\cdot 2^{m-2} + \ldots + x_0 \quad \Rightarrow \quad a^x (\mathrm{mod}\, N) = (a^{2^{m-1}})^{x_{m-1}} (a^{2^{m-2}})^{x_{m-2}} \ldots (a)^{x_0} \ (\mathrm{mod}\, N).$$

Compute a^{2^m} by squaring m times: $a \to a^2 \to a^4 \to a^8 \to \ldots$.

$$\sum_x | x\rangle \otimes | f_{a,N}(x)\rangle$$



Powers of *a* can be classically precomputed. Fourier sample (Fourier transform and measure) to determine the period Ord(*a,N*).

We can use this Order Finding algorithm to factor *N*.
Choose random *a* such that GCD(*a*,*N*)=1. Suppose *r*, the order of *a*, is even.

N divides $a^r - 1 = \left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right)$.

N does not divide $\left(a^{r/2} - 1\right)$. (If it did then order of a would be $r/2$ or less.)

Either (1) N divides $\left(a^{r/2} + 1\right)$, or (2) N has a common factor with $\left(a^{r/2} + 1\right)$, and with $\left(a^{r/2} - 1\right)$.

In case (2) we have found an (efficiently computable) factor of *N*. In case (1) we fail, and try again with another value of *a*. For at least half the possible values of *a*, we succeed! (See notes for more details.)

Public key cryptography

The discovery of Shor's algorithm in 1994 caused great excitement for two reasons.
(1) It provided evidence that BPP $\neq$ BQP.
(2) It threatens widely used public key cryptosystems (PKC).

(1) suggested that other exponential quantum speedups might be discovered.
(2) could impact personal privacy, electronic commerce, national security.

Though number theory was once regarded as the least practical subfield of mathematics, PKC changed that. We draw on number theory to construct *trap-door one-way functions*.

There are functions that are:
-- easy to compute with publicly available information.
-- easy to invert with private information.
-- hard to invert otherwise.

Using a trap-door one-way function f for private communication from Alice to Bob:
(1) Bob creates and announces the public key, keeps private key secret.
(2) Alice, using public key, evaluates f: clear text $a \rightarrow$ cipher text $b = f(a)$.
(3) Bob, using private key, inverts f: cipher text $b \rightarrow$ clear text $f^{-1}(b) = a$.

Anyone with the public key can send an encrypted message, but only Bob can decipher it.

Public key cryptography

Another application: Bob uses a trap-door one-way function *f* to sign a message.

(1) Bob creates and announces the public key, keeps private key secret.

(2) Bob, using private key, evaluates f^{-1}: message *a* $\rightarrow$ signature $b = f^{-1}(a)$.

(3) Alice, using public key, verifies signature: signature *b* $\rightarrow$ message $f(b) = a$.

Anyone with the public key can verify the signature, but only Bob can sign.

The *RSA Cryptosystem* is widely used in electronic commerce. Its security is founded on the presumed hardness of factoring.

(1) Bob chooses two large primes *p*, *q*, verifies primality, computes $N=pq$ (typically 2048 bits).

(2) Bob computes Euler phi function: phi(N)= (N-1) – (q-1) – (p-1)= (p-1)(q-1).

(3) Bob chooses random $e <$ phi(N) such that GCD(e, phi(N))=1.

(4) Bob computes $d = e^{-1}$ (mod phi(N)) --- efficient using Euclidean algorithm.

(5) Bob announces {N,e}, the public key, retains *d*, the private key.

(6) Alice encrypts: $a \rightarrow b$= f(a) = a^e (mod N) --- efficient using repeated squaring .

(7) Bob decrypts: $b \rightarrow f^{-1}(b)= b^d$ (mod N) = a.

Why it works:

$a^{\varphi(N)} = 1$ (mod N) if $GCD(a,N)=1$, because $a^{|G|}$ for any finite group G and any $a \in G$, and $|\mathbb{Z}_N^*| = \varphi(N)$.

$$\Rightarrow \quad a^{ed} = a^{1+\varphi(N)\times\text{integer}} = a.$$

Public key cryptography

How to break RSA.
-- If you can factor *N*, you can determine phi(*N*) and compute the private key.
-- It suffices to solve Order Finding. Note that Ord(*a*,*N*)=Ord(*b*,*N*)= Order(a^e,*N*).
The adversary Eve runs the Order Finding algorithm to find order of *b*, and hence of *a*.

Eve computes $\tilde{d} = e^{-1}$ (mod Ord(a)). Then Eve decrypts: $b = a^e \rightarrow (a^e)^{\tilde{d}} = a^{1+\mathrm{Ord}(a)\times\mathrm{integer}} = a.$

Thus, when quantum computers are widely available, RSA will be insecure. What then?

Three relevant time scales:

(1) How long will current systems (e.g. RSA-2048) be safe against quantum attack?
(2) How long will it take to deploy quantum-safe alternatives?
(3) How long should previously used keys remain secure?

Two solutions:
(A) Post-quantum cryptography. Conventional hardware, computational assumptions
(B) Quantum cryptography. New quantum infrastructure, "unconditional" security

Research/development focused on quantum resistance will strengthen (A).
Satellite-based communication and quantum repeaters will boost (B).
Cryptographers should be quantum savvy!

Phase Estimation

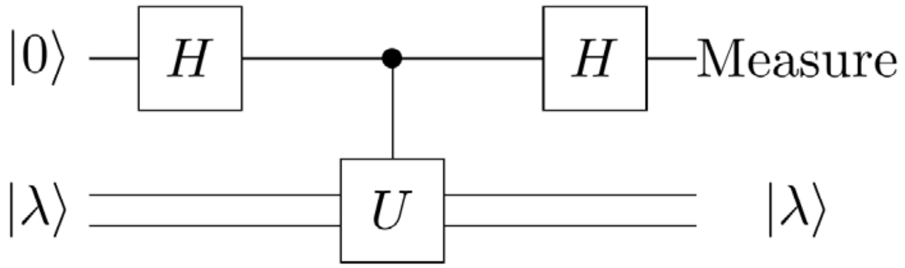
Another way to look at Period Finding and the Quantum Fourier Transform: We use a quantum computer to find the eigenvalues of a unitary operator.

$GCD(a,N)=1, \quad U_a : |x\rangle \to |ax \text{ (mod } N)\rangle, \quad$ unitary because a^{-1} exists.

$\mathrm{Ord}(a)=r \quad \Rightarrow \quad (U_a)^r = I \quad \Rightarrow \quad U_a$ eigenvalues are $\lambda_k = e^{2\pi i k/r}, k \in \{0,1,2,\ldots,r-1\}$.

Eigenstates are $|\lambda_k, x_0\rangle = \frac{1}{\sqrt{r}}\sum_{j=0}^{r-1} e^{-2\pi i j k/r}\, |a^j x_0 (\mathrm{mod}\, N)\rangle$.

Estimating eigenvalues:



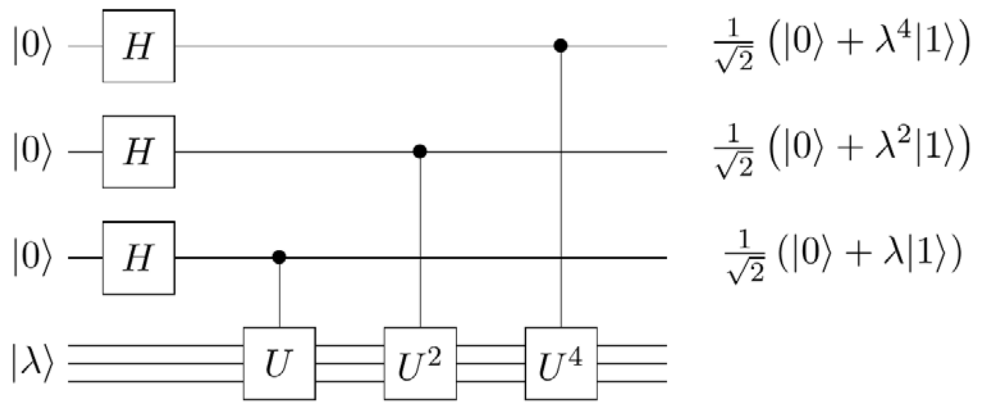
$$\frac{1}{\sqrt{2}}\big(|0\rangle+|1\rangle\big) \to \frac{1}{\sqrt{2}}\big(|0\rangle+\lambda|1\rangle\big) \quad \Rightarrow \quad \mathrm{Prob}(\pm) = \left|\frac{1}{2}(1\pm\lambda)\right|^2 = \frac{1}{2}(1\pm \mathrm{Re}\,\lambda),$$

$$\frac{1}{\sqrt{2}}\big(|0\rangle-i|1\rangle\big) \to \frac{1}{\sqrt{2}}\big(|0\rangle-i\lambda|1\rangle\big) \quad \Rightarrow \quad \mathrm{Prob}(\pm) = \left|\frac{1}{2}(1\mp i\lambda)\right|^2 = \frac{1}{2}(1\pm \mathrm{Im}\,\lambda).$$

We can perfectly distinguish eigenvalue +1 and -1, or +i and –i. Can repeat to improve the statistical error in estimate.

Phase Estimation

We can get exponential precision using the QFT, if we can compute exponentially high powers of *U* efficiently. In the case of Order Finding, we computed high powers using repeated squaring.



$|0\rangle$
$|0\rangle$
$|0\rangle$
$|\lambda\rangle$
H
H
H
U
U^2
U^4
$\frac{1}{\sqrt{2}}(|0\rangle + \lambda^4|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle + \lambda^2|1\rangle)$
$\frac{1}{\sqrt{2}}(|0\rangle + \lambda|1\rangle)$

$$\lambda = e^{2\pi i\phi} \quad \Rightarrow$$

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi\cdot 2}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi\cdot 4}|1\rangle\right) \otimes \cdots$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i\phi y}|y\rangle.$$

If $\phi = k/2^n$, then when we Fourier sample, we find k (and hence ϕ) with probability = 1.

In general, the Fourier transform is sharply peaked, so Fourier sampling determines ϕ to precision $2^{-n} \times$ O(1).

If we can compute efficiently *U* to the power *T*, we can estimate eigenvalues of *U* to precision O(1/*T*).

Prepare $\sum_{t=0}^{T}|t\rangle \otimes e^{-iEt}|E\rangle,$ Then Fourier sample the time register to estimate the frequency *E*. Estimate is limited by "time-energy uncertainty relation."