Classical RSA algorithm

We need to discuss some mathematics (number theory) first

Modulo-N arithmetic (modular arithmetic, clock arithmetic)

$$2 \equiv 9 \pmod 7$$
$$4 \times 3 \equiv 5 \pmod 7$$

"congruent" (I will also use "=" instead of "≡")

Usual operations: addition and multiplication (ring), we need only multiplication

Definition: **Order** of a is the smallest r, for which

$$a^r \equiv 1 \pmod N$$

Why important: if $f(x) = a^x \pmod N$, then r is the period of $f(x)$.

Check: $f(x+r) = a^{x+r} = a^x a^r = a^x = f(x) \pmod N$

Fermat's little theorem (simple proof, any number theory course)

If p is prime and a is not divisible by p, then

$$a^{p-1} \equiv 1 \pmod p$$

Fermat 1640 (letter, no proof)
Leibniz 1683 (unpublished)
Euler 1736 (first published proof)

(e.g., proof via the product $a(2a)(3a) \ldots ([p-1]a) = a^{p-1}(p-1)! = (p-1)! \mod p$, since all na should be different $\mod p$)

RSA mathematics

Fermat's little theorem: If p is prime and a is not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$

$\Rightarrow$ Lemma If p and q are primes and a is not divisible by p or q, then

$$a^{(p-1)(q-1)} \equiv 1 \pmod{pq}$$

Proof

$$\left.\begin{aligned} \left(a^{(p-1)}\right)^{(q-1)} &\equiv 1 \pmod{p} \\ \left(a^{(q-1)}\right)^{(p-1)} &\equiv 1 \pmod{q} \end{aligned}\right\} \Rightarrow a^{(p-1)(q-1)} - 1 \text{ is a multiple of both } p \text{ and } q, \text{ therefore multiple of } pq.$$

QED

$\Rightarrow$ Lemma If p and q are primes and s is an integer, then

$$a^{1+s(p-1)(q-1)} \equiv a \pmod{pq}$$

Note: works even if a is divisible by p or q (trivial if a multiple of pq; if only $a = kq$, then Fermat: $\left[a^{s(q-1)}\right]^{(p-1)} = 1 + np$, so $a^{s(q-1)(p-1)+1} = a + anp = a + nkqp$)

$\Rightarrow$ Theorem If $cd \equiv 1 \pmod{(p-1)(q-1)}$ and p & q are primes, then

$$a^{cd} \equiv a \pmod{pq}$$

RSA algorithm

Rivest, Shamir, Adlerman, 1977, authors from MIT

Clifford Cocks, 1973, British Intelligence, secret until 1997

Alice ⇄ Bob (public key →, message ←) (in Mermin's book roles of Alice and Bob are exchanged)

Alice Pick large primes p and q, calculate $N = pq$

Pick $c < N$ [coprime with $(p-1)(q-1)$]

Find d, for which $cd \equiv 1 \pmod{(p-1)(q-1)}$

(easy to find d using Euclidean algorithm for c and $(p-1)(q-1)$)

Public key: N and c

Private key: N and d

Bob Wants to send message a $(a < N)$

Encoding: $a \rightarrow \tilde{a} = a^c \pmod{N}$

Alice Decoding: $\tilde{a}^d \pmod{N} = a^{cd} \pmod{N} = a$

RSA algorithm (cont.)

Remarks

- Typically $N \sim 2048 - 4096$ bits long

- Computation of $a^c \pmod N$ and $\tilde{a}^d \pmod N$ is fast:
  $a \to a^2 \to a^4 \to a^8 \to \ldots$, then products (all $\bmod N$)

- Eve knows N. If she can factor $N = pq$, then she can do the same as Alice, so she can decode. This is why factoring is so important.

- N can be factored via finding the period of the function $f(x) = a^x \pmod N$, where a is any number (will discuss in more detail later).
  Idea: if $a^r \equiv 1 \pmod N$ and r is even, then
  $\left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right) \equiv 0 \pmod N$

- RSA can be also broken directly with a period-finding algorithm.
  $\tilde{a}, \tilde{a}^2, \tilde{a}^3, \ldots \tilde{a}^r = 1, \tilde{a}^{r+1} = \tilde{a} \pmod{\mathrm{N}}$ (if $\tilde{a}$ is not coprime with N, then factor immediately). Then $a^r \equiv 1 \pmod N$ also (because subgroups $\tilde{a}^k$ and a^k coincide since $a^c \equiv \tilde{a}$, and $\tilde{a}^d \equiv a$, so the same order.)
  Then if we find d' so that $cd' \equiv 1 \pmod r$, then
  $\tilde{a}^{d'} \equiv a^{cd'} \equiv a^{1+mr} = a\,(a^r)^m = a$, so direct decoding.

Classical algorithm for factoring via period finding

$N = pq$ can be factored via period of $f(x) = a^x \pmod N$

1. Pick a random number a ($a < N$).
   Check that coprime with N (if not, then great luck!).

2. Find smallest r, for which $a^r \equiv 1 \pmod N$ (i.e., r is the order of a).

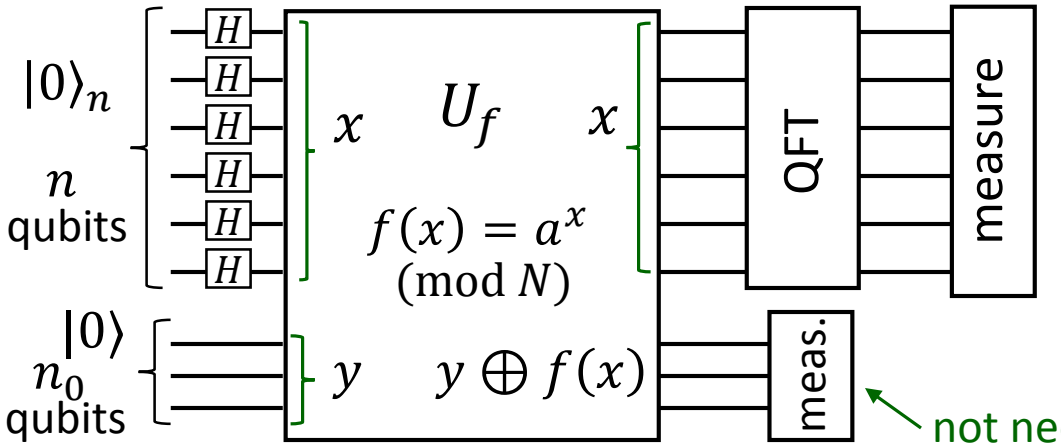
3. If r is odd, choose another a and repeat (go back to Step 1).
   Probability of going back is ~50%.

4. If r is even, then $\left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right) = a^r - 1 \equiv 0 \pmod N$.
   $a^{r/2} - 1$ cannot be $0 \pmod N$, since r is the smallest period.
   If $a^{r/2} + 1 \equiv 0 \pmod N$, choose another a and repeat (go back to Step 1; this is very rare).

5. Since $N = pq$ and p & q are primes, then $a^{r/2} - 1$ is a multiple of p, and $a^{r/2} + 1$ is a multiple of q (or vice versa).
   Find the greatest common divisor (GCD) of N and $a^{r/2} \pm 1$, they will be p and q.

Remarks
- If p and q are not prime, then similar algorithm.
- If r is not the smallest period, then check that $a^{r/2} - 1$ is not 0 (mod N), otherwise choose another a (very rare)

General idea of period finding by a QC (Shor's algorithm)



not needed, but easier to think

$f(x) = a^x \pmod N$

N has n_0 bits

Output register has n_0 qubits

Input register has $n \geq 2n_0$ qubits

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle_n|0\rangle_{n_0} \longrightarrow \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle_n|f(x)\rangle_{n_0}$$

After meas. of output register, the input reg. is $|\psi\rangle_n = \frac{1}{\sqrt{m}}\sum_{k=0}^{m-1}|x_0 + kr\rangle_n$

where r is the period of $f(x)$ (i.e., order of a), $m = \text{int}[2^n/r]$ or $\text{int}[2^n/r]+1$

$r < N < 2^{n_0}$, so $m > 2^{n_0}$ (very many states in superposition)

Idea: Input register state is periodic (r) $\Rightarrow$ Fourier transform finds this period

Key: Quantum Fourier transform (QFT) can be done very efficiently

For $M \sim 2^n$, usual Fourier transform needs $\sim M^2 \sim (2^n)^2$ operations,

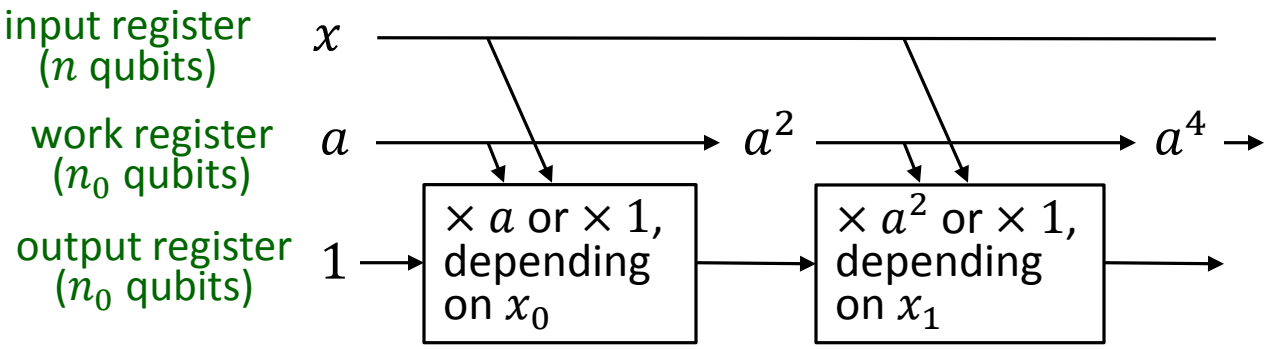
Fast Fourier Transform (FFT) needs $\sim M \sim 2^n$ operations (actually $n2^n$),

QFT needs $\sim (\log M)^2 \sim n^2$ operations. (Calculation of $f(x)$ needs $\sim n^3$ operations.)

Calculation of $f(x) = a^x \pmod N$

Fast classical algorithm $\Rightarrow$ quantum algorithm of the same complexity

Prepare a, a^2, a^4, a^8, . . . $(\mathrm{mod}\ N)$, then multiply some of them, depending on the corresponding bits of $x = x_{n-1} \ldots x_1 x_0$



Complexity: n steps, each contains multiplication $(\mathrm{mod}\ N)$ requiring $\sim n^2$ steps, so overall $\sim n^3$ steps

$(n_0 \sim n)$

By the way, in this algorithm the work register remains unentangled with input and output registers, so no "global" garbage collection is needed (garbage collection at each step is still necessary)

Quantum Fourier Transform (QFT)

Discrete Fourier transform (DFT)

$$x = 0, 1, 2, \ldots M-1 \qquad\qquad g(x) \to \tilde{g}(x)$$

$$\tilde{g}(x) = \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{2\pi i x y / M} g(y)$$

Inverse DFT: the same with $i \to -i$

In QC, $M = 2^n$ (n qubits), and we do discrete Fourier transform of amplitudes:

$$\sum_{x=0}^{2^n-1} g(x) \, |x\rangle \overset{U_{QFT}}{\longrightarrow} \sum_{x=0}^{2^n-1} \tilde{g}(x) \, |x\rangle$$

Therefore $\boxed{U_{QFT} \, |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i x y / 2^n} |y\rangle}$

- Check that unitary. For basis vectors $|x_l\rangle$ and $|x_m\rangle$, the inner product after QFT is $\langle x_l U_{QFT}^\dagger | U_{QFT} x_m \rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} e^{2\pi i (-x_l + x_m) y / 2^n} \langle y | y \rangle = \frac{1}{2^n} 2^n \delta_{lm} = \delta_{lm}$.
  So, the orthonormal basis is transformed into an orthonormal basis $\Rightarrow$ unitary.

- Somewhat similar to n-fold Hadamard: transforms each basis vector into equal-weight superposition of all basis vectors (but instead of ± 1 for Hadamard, many phases in QFT)

Quantum Fourier Transform (cont.)

$$U_{QFT}\,|x\rangle = \frac{1}{\sqrt{2^n}}\ \sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n}|y\rangle$$

A very simple quantum circuit exits for QFT

$$U_{QFT}\,|x\rangle = \frac{1}{\sqrt{2^n}}\ \sum_{y_{n-1},\dots y_0} e^{2\pi i x(y_{n-1}2^{n-1}+y_{n-2}2^{n-2}+\dots+y_0 2^0)/2^n}|y_{n-1}\rangle|y_{n-2}\rangle\dots|y_0\rangle$$

$$= \frac{1}{\sqrt{2^n}}\left(|0\rangle + |1\rangle e^{2\pi i x 2^{n-1}/2^n}\right)\left(|0\rangle + |1\rangle e^{2\pi i x 2^{n-2}/2^n}\right)\dots\left(|0\rangle + |1\rangle e^{2\pi i x 2^0/2^n}\right)$$

For $x = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \dots x_0 2^0$, many digits are not important

$$U_{QFT}\,|x\rangle = \frac{|0\rangle + |1\rangle\, e^{2\pi i \frac{x_0}{2}}}{\sqrt{2}}\ \frac{|0\rangle + |1\rangle\, e^{2\pi i\left(\frac{x_1}{2}+\frac{x_0}{2^2}\right)}}{\sqrt{2}}\ \dots \frac{|0\rangle + |1\rangle\, e^{2\pi i\left(\frac{x_{n-1}}{2}+\frac{x_{n-2}}{2^2}+\dots+\frac{x_0}{2^n}\right)}}{\sqrt{2}}$$

First (most significant) qubit: $\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\, e^{2\pi i \frac{x_0}{2}}\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\,(-1)^{x_0}\right) = H|x_0\rangle$

(only in computational basis)

So, if we use <u>reverse order</u> (most significant ↔ least significant), then the only necessary operation is H acting on qubit $|x_0\rangle$.

Second qubit: needs H acting on $|x_1\rangle$ and also $\begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi\, i/2^2) \end{pmatrix}$ if $x_0 = 1$.

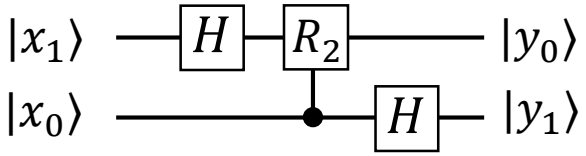
Quantum Fourier Transform (cont.)

$$U_{QFT}\,|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{2\pi i xy/2^n}|y\rangle$$

$$= \frac{|0\rangle + |1\rangle\, e^{2\pi i \frac{x_0}{2}}}{\sqrt{2}}\,\frac{|0\rangle + |1\rangle\, e^{2\pi i\left(\frac{x_1}{2}+\frac{x_0}{2^2}\right)}}{\sqrt{2}}\cdots\frac{|0\rangle + |1\rangle\, e^{2\pi i\left(\frac{x_{n-1}}{2}+\frac{x_{n-2}}{2^2}+\ldots+\frac{x_0}{2^n}\right)}}{\sqrt{2}}$$

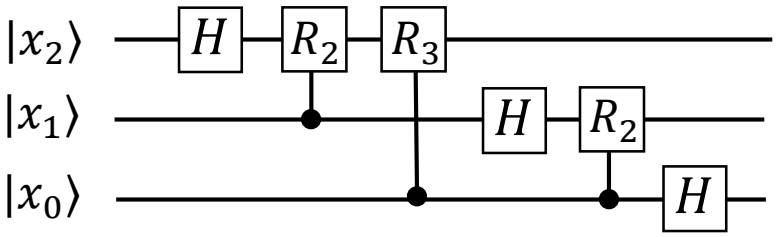
Let us introduce rotation operator $R_k \equiv \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi\, i/2^k) \end{pmatrix}$ (Mermin: $R_k = V_{k-1}$)

Two qubits



(reverse order)

Three qubits



$$|y_0\rangle = \left(|0\rangle + |1\rangle\, e^{2\pi i\left(\frac{x_2}{2}+\frac{x_1}{2^2}+\frac{x_0}{2^3}\right)}\right)/\sqrt{2}$$

$$|y_1\rangle = \left(|0\rangle + |1\rangle\, e^{2\pi i\left(\frac{x_1}{2}+\frac{x_0}{2^2}\right)}\right)/\sqrt{2}$$

$$|y_2\rangle = \left(|0\rangle + |1\rangle\, e^{2\pi i\left(\frac{x_0}{2}\right)}\right)/\sqrt{2}$$

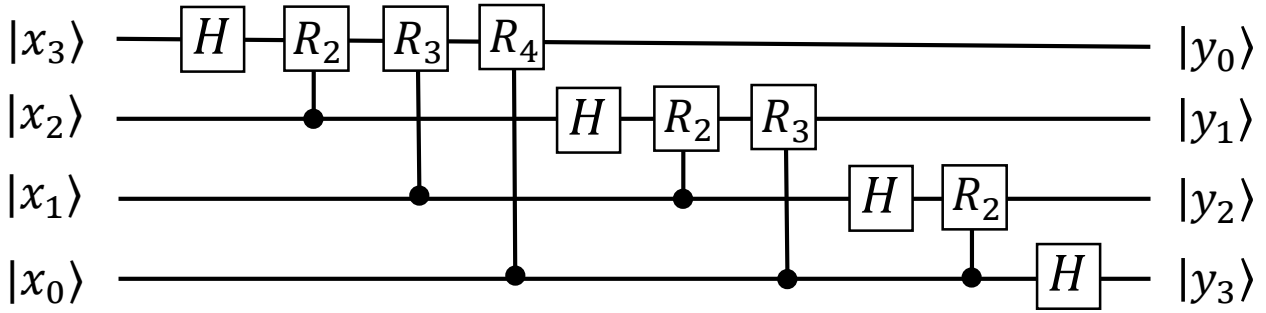
again, output order is reversed

Quantum Fourier Transform (cont.)

$$U_{QFT}\,|x\rangle = \frac{|0\rangle + |1\rangle\, e^{2\pi i \frac{x_0}{2}}}{\sqrt{2}} \frac{|0\rangle + |1\rangle\, e^{2\pi i (\frac{x_1}{2}+\frac{x_0}{2^2})}}{\sqrt{2}} \cdots \frac{|0\rangle + |1\rangle\, e^{2\pi i (\frac{x_{n-1}}{2}+\frac{x_{n-2}}{2^2}+\ldots+\frac{x_0}{2^n})}}{\sqrt{2}}$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n} |y\rangle \qquad R_k = \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi\, i/2^k) \end{pmatrix}$$

Four qubits

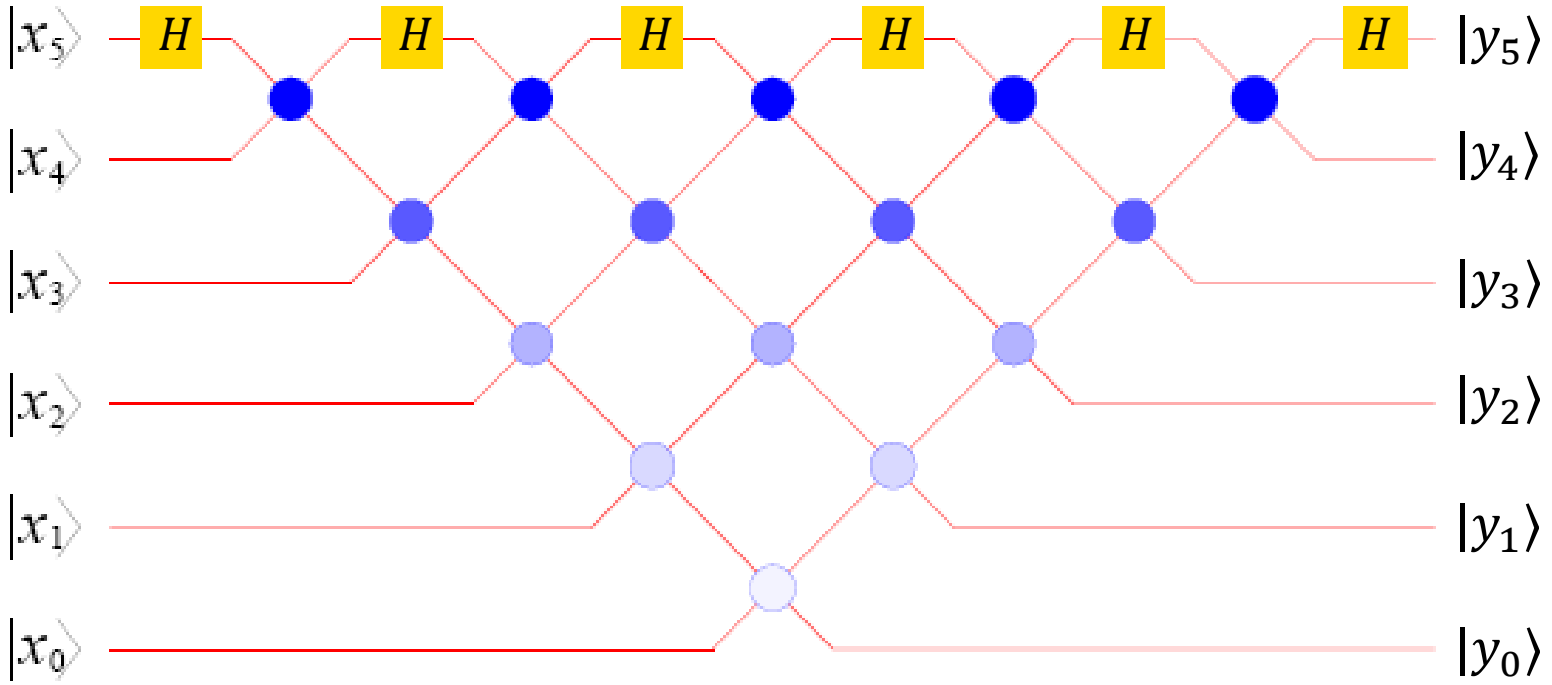


reversed order

Similar for n qubits: need n Hadamard gates and $n(n-1)/2$ controlled-*R* gates.
Each c-*R* gate can be realized with 2 CNOTs, so $\sim n^2$ CNOTs.
(With superconducting qubits, c-R gate can be realized directly.)

c-*R* gates with extreme precision ($\sim 2^{-n}$) are actually not needed. Crude precision is sufficient (will discuss later), so gates c-R_k with $k > 20$ are not needed. Then only $\sim 20n$ c-*R* gates are needed.

Another representation of the same circuit for QFT

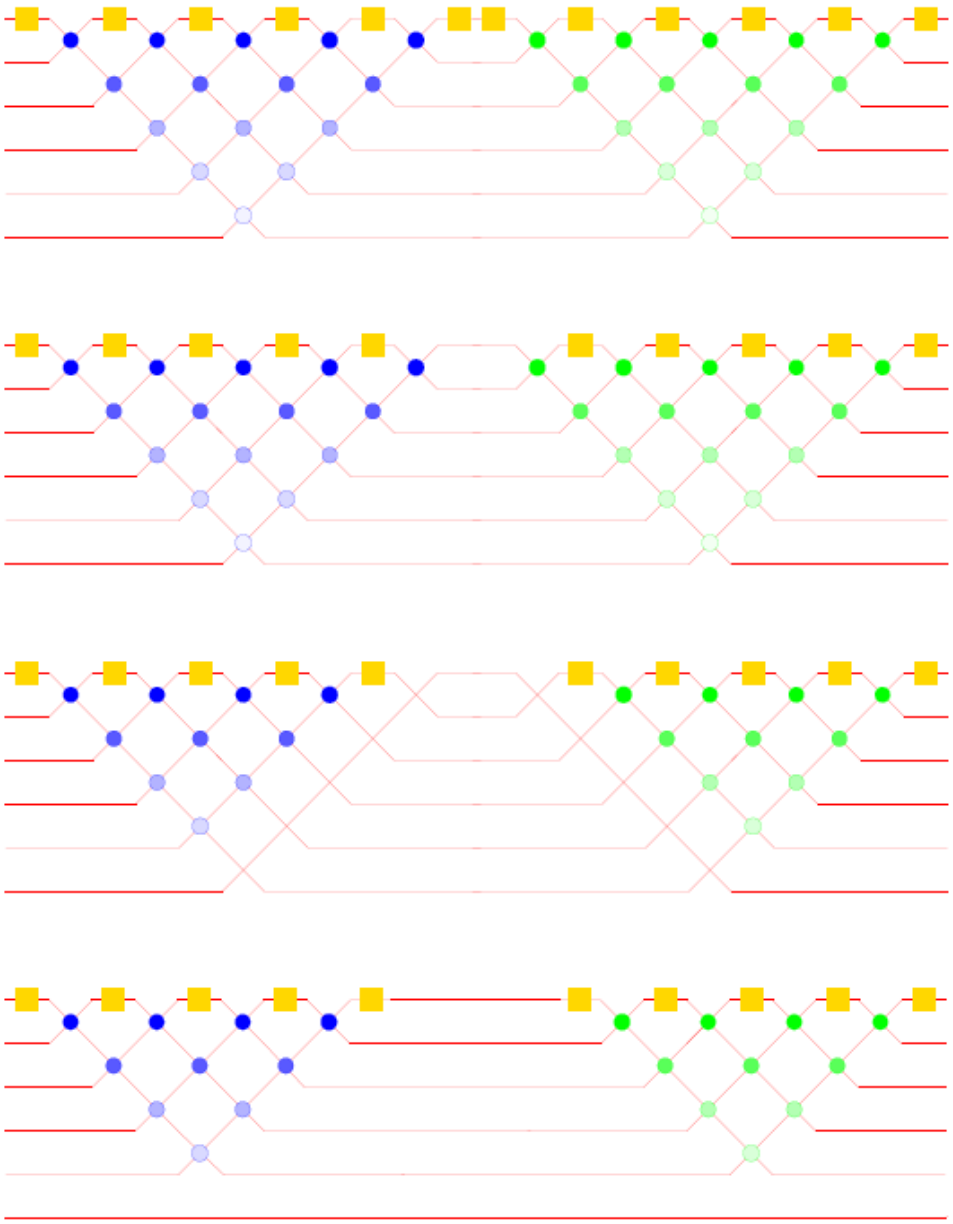


Symmetry of c-R gates and reversed order are naturally represented

Inverse QFT: time-reverse the sequence and conjugate gates

($H^\dagger = H$, so only replace c-$R_k \rightarrow$ c-$R_k^\dagger$)

Inverse QFT in this representation

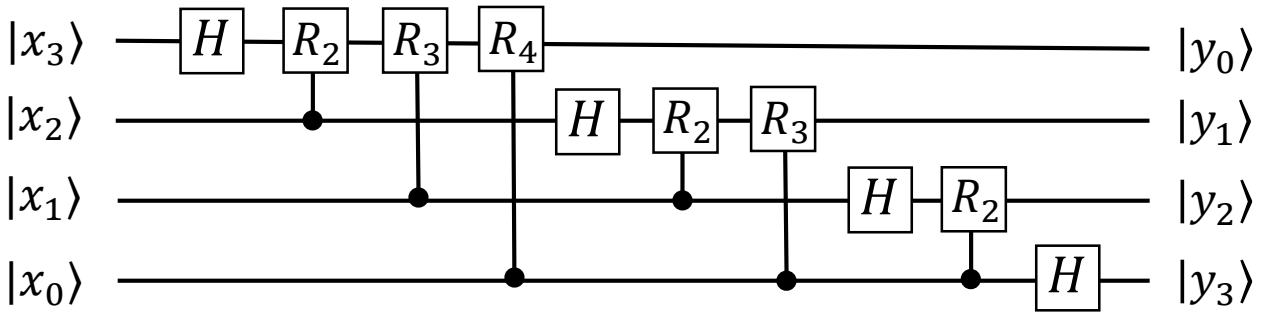


yellow: H

blue: c-R_k

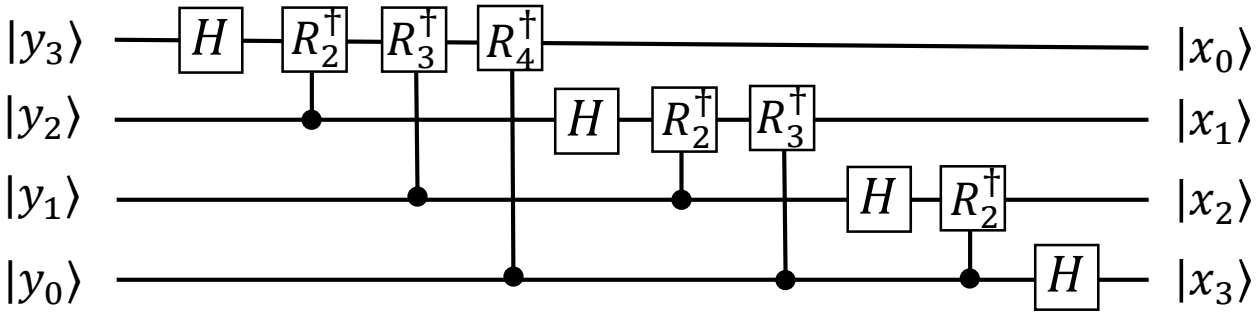
green: c-$R_k^\dagger$

Inverse QFT using the first circuit



QFT

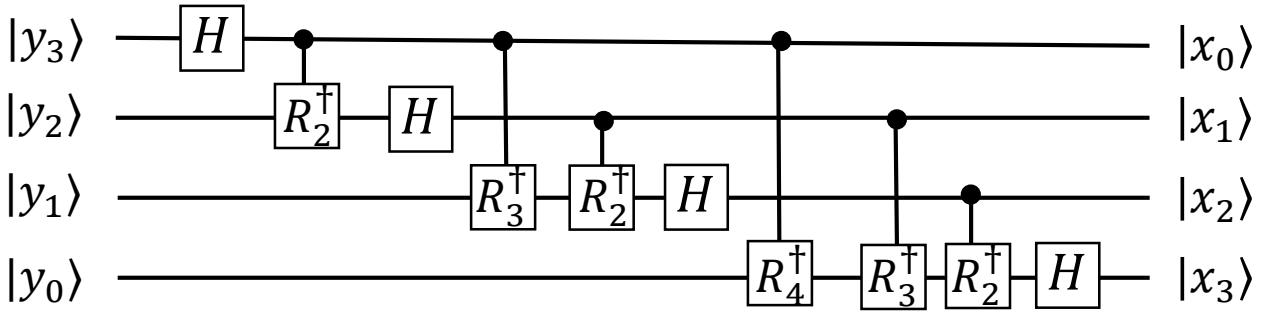
Inverse QFT: $i \to -i$, so we would expect



QFT^{-1}

On the other hand, we know that for inverse, the circuit should be time-reversed and gates should be conjugated.

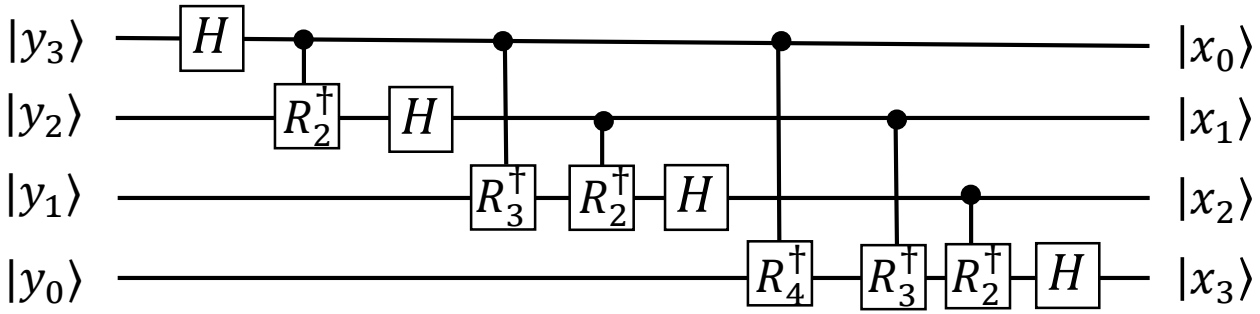
Does not look the same! But actually is.



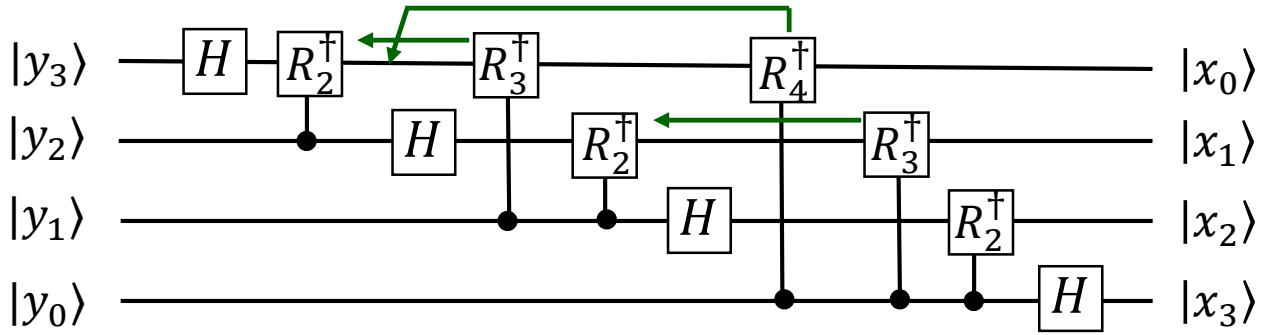
QFT^{-1}

use symmetry of c-R_k, then shift gates

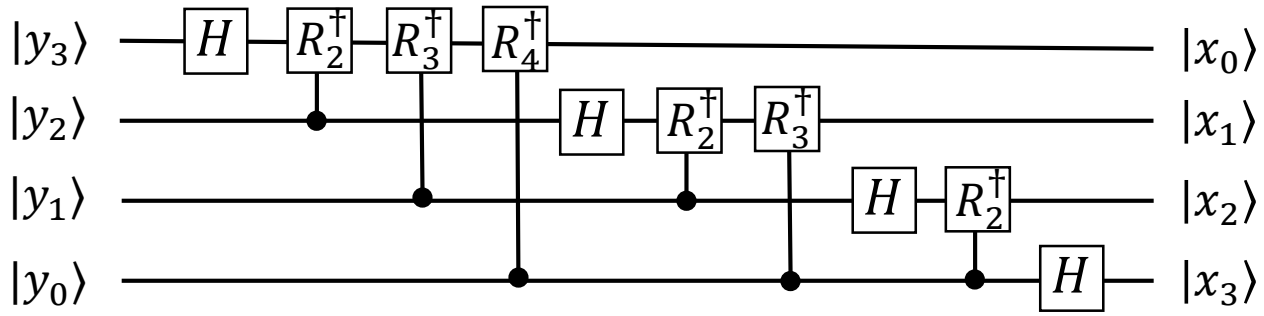
Inverse QFT (cont.)



use symmetry of c-R_k



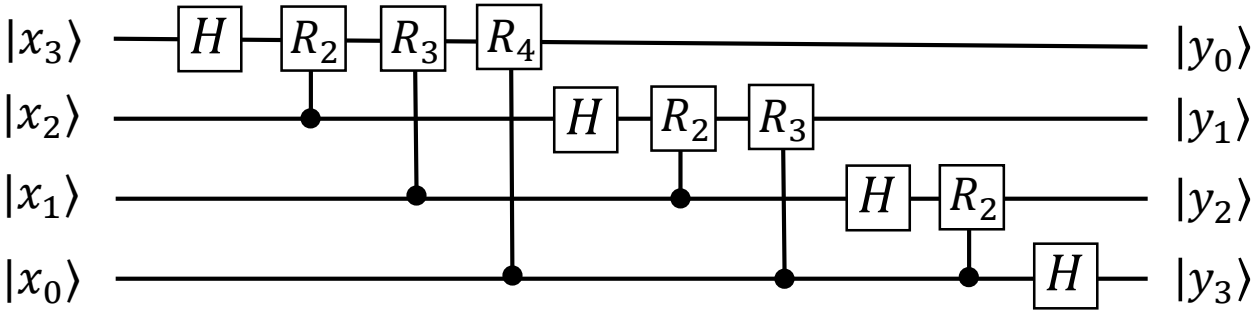
shift some gates to the left



Measurement-based realization of QFT

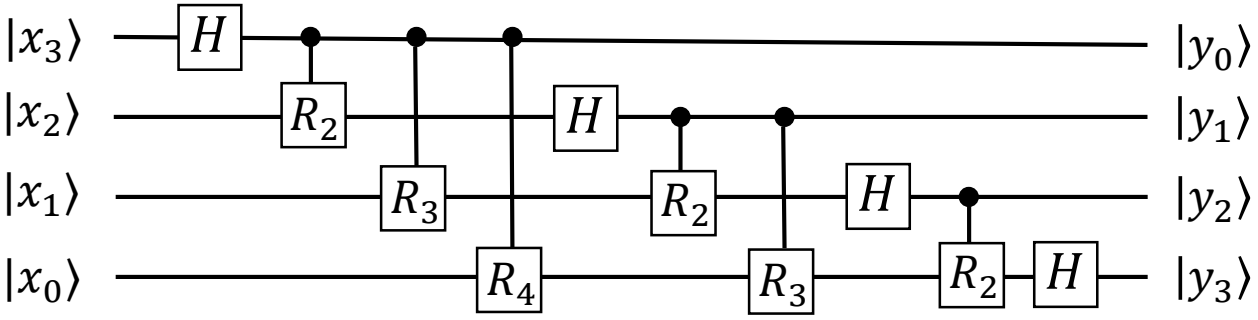
In Shor's algorithm, all qubits are measured after QFT. In this case QFT can be realized with classically-controlled R_k gates.

Usual QFT



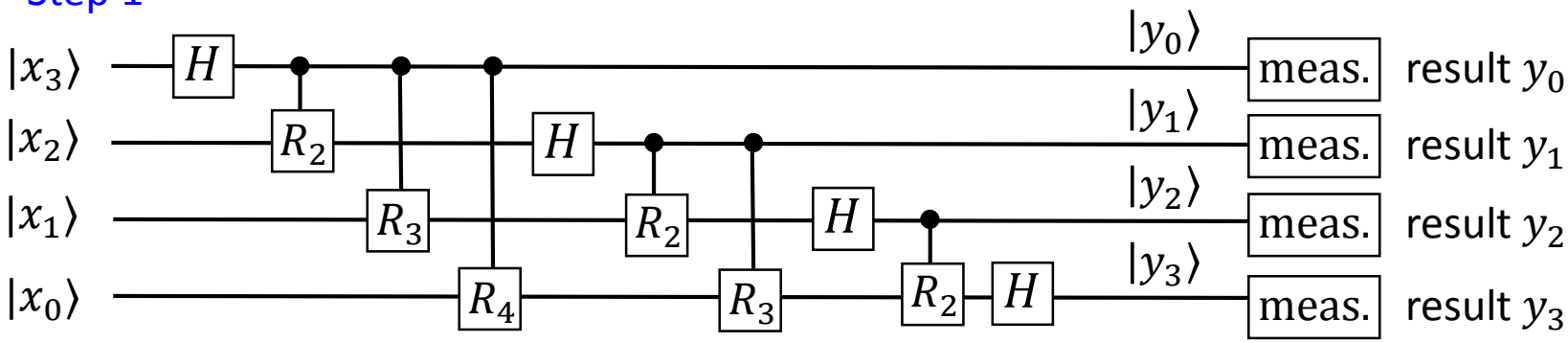
Step 1

Since c-R_k gates are symmetric, exchange control and target



Measurement-based realization of QFT (cont.)

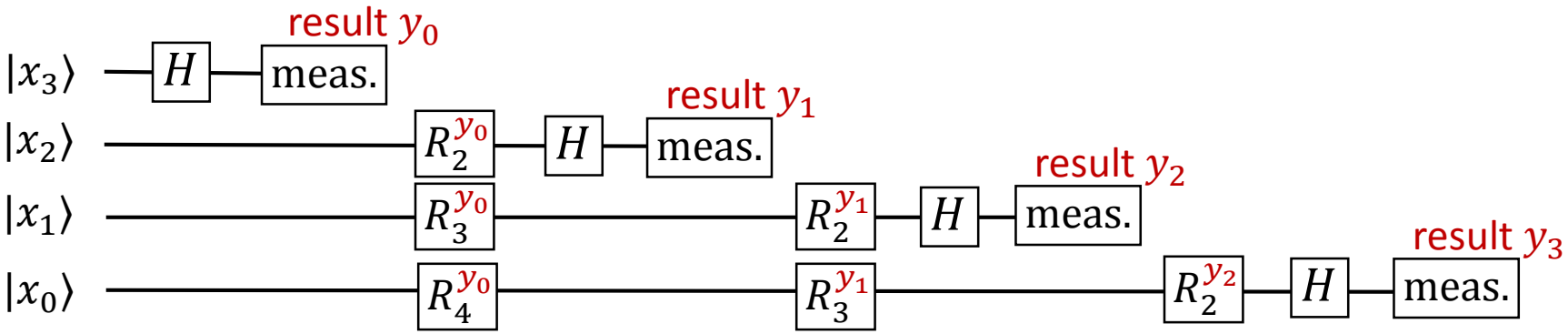
Step 1



Step 2

Measure and control classically

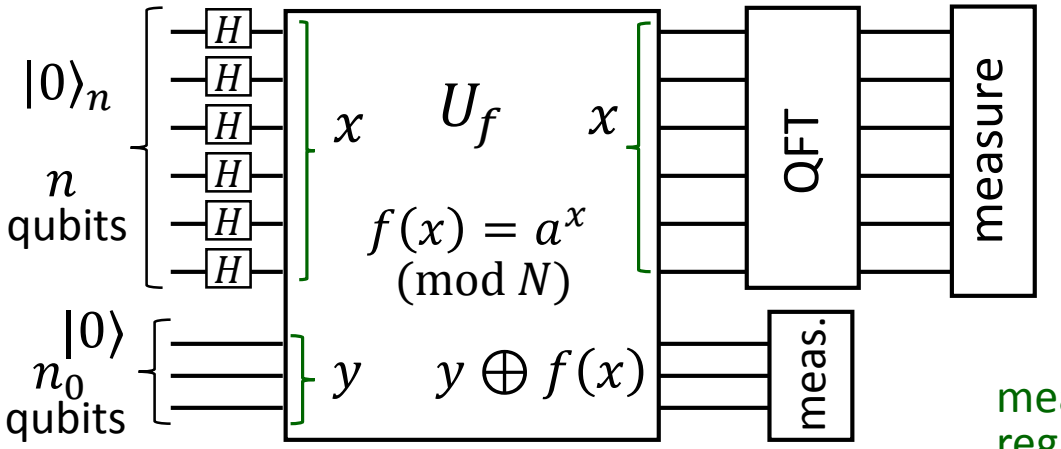
$$R_k \equiv \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi\, i/2^k) \end{pmatrix}$$



Because of “spooky action”, measurement acts back in time, so we can exchange in time measurement and control

So far we assume that gates are perfect (it is not possible experimentally for R_k with exponentially small angles). We will discuss later that precision is not a problem.

Back to Shor's algorithm (period finding)



measure second register, result $f(x_0)$

period we want to find

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle_n|0\rangle_{n_0} \xrightarrow{U_f} \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle_n|f(x)\rangle_{n_0} \longrightarrow \frac{1}{\sqrt{m}}\sum_{k=0}^{m-1}|x_0+kr\rangle_n \longrightarrow$$

$$m = \text{int}[2^n/r]$$

$$\xrightarrow{U_{\text{QFT}}} \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}\frac{1}{\sqrt{m}}\sum_{k=0}^{m-1} e^{2\pi i\,(x_0+kr)y/2^n}|y\rangle_n$$

$$= \frac{1}{\sqrt{2^n}}\frac{1}{\sqrt{m}}\sum_{y=0}^{2^n-1} e^{2\pi i x_0 y/2^n}\sum_{k=0}^{m-1} e^{2\pi i\,kry/2^n}|y\rangle_n$$

Measure first register, probability of result y is

$$p(y) = |\psi(y)|^2 = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i\,kry/2^n}\right|^2$$

x_0 is not important, just a phase factor

No more QM, let us see how result is related to r

Shor's algorithm (cont.)

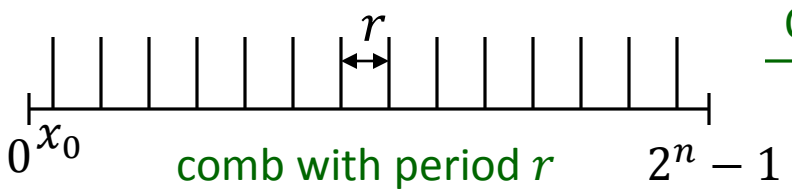
$$p(y) = |\psi(y)|^2 = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i\, kry/2^n}\right|^2$$

integer

Significant $p(y)$ only if all terms are in phase: $y \approx \frac{2^n}{r}\, j$

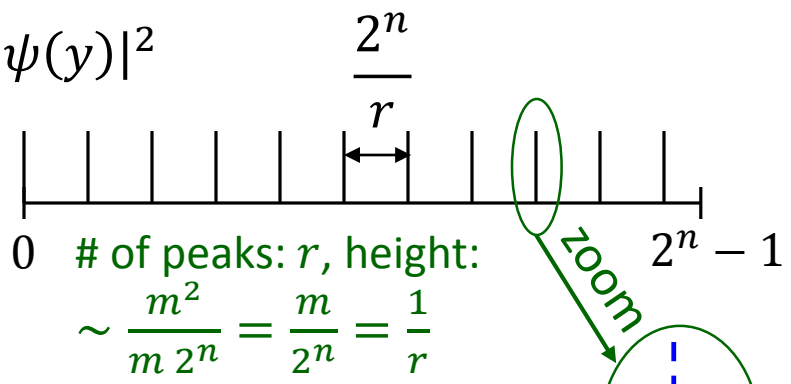
Understanding via Fourier transform

$|\psi\rangle = \frac{1}{\sqrt{m}}\sum_{k=0}^{m-1} |x_0 + kr\rangle_n$



QFT

$|\psi(y)|^2$



Measurement randomly picks one of the peaks of $p(y)$, while we need r.

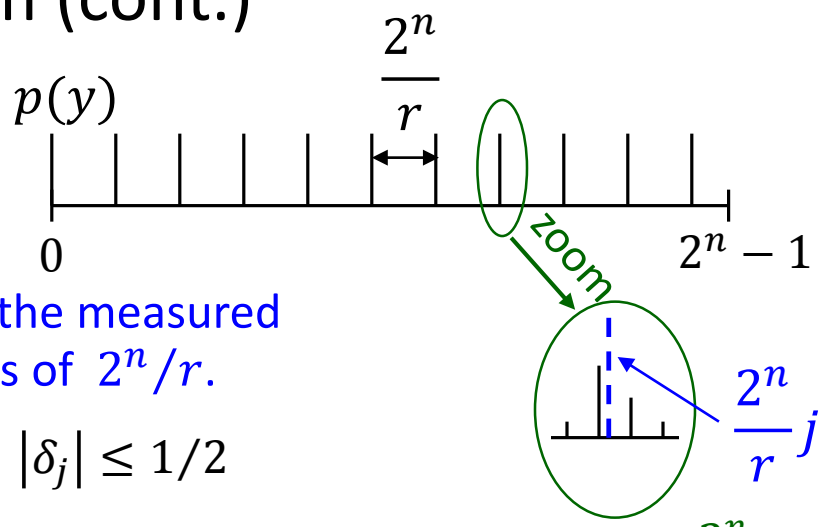
Two steps next:

1) Show that with a significant probability (>40%) the measured number is the closest (<1/2) to one of multiples of $2^n/r$.

2) Show that in this case, from the measured number we can obtain r.

Shor's algorithm (cont.)

$$p(y) = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i\, kry/2^n}\right|^2$$



1) Show that with a significant probability (>40%) the measured number is the closest (<1/2) to one of multiples of $2^n/r$.

Denote the closest integer as $y_j = j\,2^n/r + \delta_j\,,\ |\delta_j| \le 1/2$

Sum geometric series for $p(y_j)$

$$p(y_j) = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i k r \delta_j/2^n}\right|^2 = \frac{1}{2^n m}\left|\frac{e^{2\pi i m r \delta_j/2^n} - 1}{e^{2\pi i r \delta_j/2^n} - 1}\right|^2 = \frac{1}{2^n m}\,\frac{\sin^2(\pi m r\,\delta_j/2^n)}{\sin^2(\pi r\,\delta_j/2^n)}$$

$$\approx \underbrace{\frac{1}{2^n m}}_{\approx 4^n/r}\,\frac{\sin^2(\pi\delta_j)}{\sin^2(\underbrace{\pi r\,\delta_j/2^n}_{\text{very small, } r < 2^{n_0} \ll 2^n})} \approx \frac{1}{r}\left(\frac{\sin(\pi\delta_j)}{\pi\,\delta_j}\right)^2 \ge \frac{1}{r}\,\frac{4}{\pi^2}$$

at $\delta_j = \pm 1/2$

$\approx r$ peaks ($j2^n/r,\ j = 1, 2,\ \ldots r-1$), so total probability that measured result is within $1/2$ from $j2^n/r$ is $\ge 4/\pi^2 > 40\%$. Not always but quite likely.

Actually, if try both neighbors, then probability to be within $1/2$ from $j2^n/r$ is $> 80\%$, if try 4 closest neighbors, then $> 90\%$.

Shor's algorithm (cont.)

2) How to find period r from $y = j\,2^n/r + \delta$, where $|\delta| \le 1/2$

Rewrite $\left| \frac{y}{2^n} - \frac{j}{r} \right| \le \frac{1}{2^{n+1}}$

we know

want to find

Remember $r < N < 2^{n_0}$

integer to factor

# of bits in N

n is a parameter we can choose. For large enough n, the result $y/2^n$ will be very close to the rational number j/r.

Rational numbers with denominators $< N$ are not closer to each other than $1/N^2$
(because $|a/b - c/d| \ge 1/bd$)

So, if $\frac{1}{2^{n+1}} \le \frac{1}{2N^2}$, then the closest to $y/2^n$ rational number with denominator $\le N$ is j/r.

This is why we need $n \ge 2n_0$.

How to find j/r: continued fractions

$$\frac{y}{2^n} = \frac{1}{z_0 + \frac{1}{z_1 + \frac{1}{z_2 + \ldots}}} ,$$

This expansion will go through j/r

Theorem: If x is an estimate of j/r, $|x - j/r| \le 1/(2r^2)$, then continued fractions go through j/r (proven in N-C book, not a very short proof)

Continued fractions is a fast classical algorithm, $O(n_0^3)$ operations

Shor's algorithm (cont.)

Finding period r

So, we will find j/r with a significant probability ($> 40\%$).

It is still possible that we will not find correct r if j and r have common divisors. Then we will find a divisor of r instead of r itself.

However, the probability of finding r (not its divisor) is $\geq 50\%$, and if it is not r, then it is most likely $r/2$ or $r/3$ (not large denominator). So, after finding r_0, we can try r_0, $2r_0$, $3r_0$, etc.

It is important that it is easy to check classically if kr_0 is a period of $f(x)$ or not.

If the procedure is unsuccessful, we can run the algorithm again (with the same a). If find another divider of r, we can calculate Least Common Multiple (LCM); most likely if will be r.

Still possible that $y/2^n$ was not the closest j/r, so need several trials.

So, $\sim 3 - 10$ runs of the quantum algorithm will give us the period r.

Required precision of gates c-R_k in QFT

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi\, i/2^k) \end{pmatrix}$$

For $k > 10$ it is very difficult to realize c-R_k accurately,
for $k > 20$ practically impossible.

Is this precision (very small angles) really necessary? No!

General idea

If a gate is imprecise, then $|\psi\rangle \to |\psi'\rangle$. But if the imprecision is not too big, then the states $|\psi\rangle$ and $|\psi'\rangle$ are still close, $|\langle\psi|\psi'\rangle|^2 = 1 - \varepsilon$ with $\varepsilon \ll 1$.

Then they are not well-distinguishable (independently of what we measure).

So, probability of measuring what we want does not change much.

In some sense, the operation is digital, and therefore insensitive to small analog errors.

Required precision of gates c-R_k in QFT (cont.)

Estimate of phase accuracy needed for QFT

Ideally, $p(y) = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i k r y/2^n}\right|^2$

Suppose there are phase errors

$$p_\varphi(y) = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i k r y/2^n}\, \underbrace{e^{i\varphi_k(y)}}_{\approx 1 + i\varphi_k(y)}\right|^2$$

can depend on both k and y

Assume $|\varphi_k(y)| \le \varphi \ll 1$

As before, $y_j = j\,2^n/r + \delta_j$ with $|\delta_j| \le 1/2$

$$p_\varphi(y_j) \approx \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i k r \delta_j/2^n}\left(1 + i\varphi_{k,j}\right)\right|^2 \approx$$

for $y = y_j$; in linear order

$$\approx p(y_j) + \frac{2}{2^n m}\mathrm{Re}\left[\left(\sum_{k=0}^{m-1} e^{2\pi i k r \delta_j/2^n}\, i\varphi_{k,j}\right)\left(\sum_{k'=0}^{m-1} e^{-2\pi i k' r \delta_j/2^n}\right)\right]$$

ideal

Even in the worst case: $\left|\sum_{k=0}^{m-1} e^{2\pi i k r \delta_j/2^n}\, i\varphi_{k,j}\right| \le m\varphi\,,\ \left|\sum_{k'=0}^{m-1} e^{-2\pi i k' r \delta_j/2^n}\right| \le m$

So difference is limited: $|p_\varphi(y_j) - p(y_j)| \le \frac{2}{2^n m}\, m\varphi\, m = \frac{2m}{2^n}\varphi \approx \frac{2}{r}\varphi$

Total difference $\le r\left|p_\varphi(y_j) - p(y_j)\right| \le 2\varphi \ll 1$ Small!

Required precision of gates c-R_k in QFT (cont.)

$$p(y) = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i k r y/2^n}\right|^2$$ ideally

$$p_\varphi(y) = \frac{1}{2^n m}\left|\sum_{k=0}^{m-1} e^{2\pi i k r y/2^n}\, e^{i\varphi_k(y)}\right|^2$$ with phase errors $|\varphi_k(y)| \le \varphi \ll 1$

Total probability difference $\le r\left|p_\varphi(y_j) - p(y_j)\right| \le 2\varphi$

Therefore, the probability of success (i.e. the measured y is the closest integer to $j\ 2^n/r$) is not $\ge 40\%$, but $\ge 40\% - 2\varphi$.

Therefore the precision $\varphi \sim 10\%$ is sufficient! (digital computation)

We still cannot say that all gates with 3% accuracy is OK, because many gates for each "wire"

Inaccuracy scales (at most) linearly with the number of gates.

In QFT, there are $\sim n$ gates R_k.

The gates R_k can be completely neglected if $n\ 2\pi\ 2^{-k} < 0.1$

Therefore $k_{max} \sim \log_2(n) + 6 \sim 20$ is sufficient

Then the number of gates in QFT is not $\sim n^2$ but only $\sim n\log(n)$

Precision of gates (more general discussion)

Introduce operator norm

$$\|\hat{O}\| = \sup_{|\psi\rangle\neq 0} \frac{\|\hat{O}|\psi\rangle\|}{\| |\psi\rangle \|} = \sup_{|\psi\rangle\neq 0} \sqrt{\frac{\langle\psi|\hat{O}^\dagger\hat{O}|\psi\rangle}{\langle\psi|\psi\rangle}}$$

max

It is really a norm (satisfies triangle inequality)

Imprecision of a gate

Suppose a unitary U is replaced with a slightly imprecise unitary U'. The imprecision can be characterized by the norm of the difference: $\Delta = \|U - U'\|$.

Then for an imprecise sequence of gates (composition of operations),
$U_k \dots U_2 U_1 \longrightarrow U'_k \dots U'_2 U'_1$, we can show $\Delta \leq \sum_i \Delta_i$

The proof is step-by-step, using triangle inequality and norm-preservation by a unitary

$$U_2U_1|\psi\rangle - U'_2U'_1|\psi\rangle = (U_2U_1|\psi\rangle - U'_2U_1|\psi\rangle) + (U'_2U_1|\psi\rangle - U'_2U'_1|\psi\rangle) =$$
$$= (U_2 - U'_2)U_1|\psi\rangle - U'_2(U_1 - U'_1)|\psi\rangle$$

Therefore $\|U_2U_1 - U'_2U'_1\| \leq \|U_2 - U'_2\| + \|U_1 - U'_1\|$

So, we proved that the imprecision Δ accumulates at most linearly with the number of gates

Precision of gates (cont.)

We proved that the imprecision Δ accumulates at most linearly with the number of gates.

Two more important properties:

For an overall imprecision Δ, the difference in the probability of obtaining a certain result for a measurement is less than 2Δ (simple proof in N-C book, Sec. 4.5.3).

If a 1-qubit or 2-qubit gate U has imprecision Δ, then the same imprecision for this gate acting on many-qubit state (i.e., gate $U \otimes \hat{1}$).

Proof (for a 2-qubit gate) A multi-qubit entangled state can always be represented as

$$|\Psi\rangle = \alpha_{00}|00\rangle|\Phi_{00}\rangle + \alpha_{01}|01\rangle|\Phi_{01}\rangle + \alpha_{10}|10\rangle|\Phi_{10}\rangle + \alpha_{11}|11\rangle|\Phi_{11}\rangle,$$

where $|\Phi_{ij}\rangle$ are normalized states of other qubits, $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

A gate U acts only on α_{ij}, an imprecise U' produces $\alpha_{ij,\text{in}} \to \alpha'_{ij}$ instead of $\alpha_{ij,\text{in}} \to \alpha_{ij}$.

Then $\|U - U'\| = \max \|\Psi' - \Psi\| = \max \|(\alpha'_{00} - \alpha_{00})|00\rangle|\Phi_{00}\rangle +$

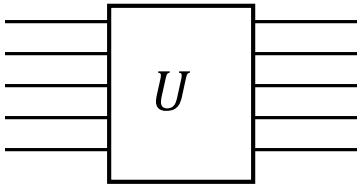
$$+(\alpha'_{01} - \alpha_{01})|01\rangle|\Phi_{01}\rangle + (\alpha'_{10} - \alpha_{10})|10\rangle|\Phi_{10}\rangle + (\alpha'_{11} - \alpha_{11})|11\rangle|\Phi_{11}\rangle\| =$$

$$= \max \sqrt{(\alpha'_{00} - \alpha_{00})^2 + (\alpha'_{01} - \alpha_{01})^2 + (\alpha'_{10} - \alpha_{10})^2 + (\alpha'_{11} - \alpha_{11})^2},$$

which is the same as when this gate acts only on two qubits. QED

Phase estimation algorithm (Kitaev)

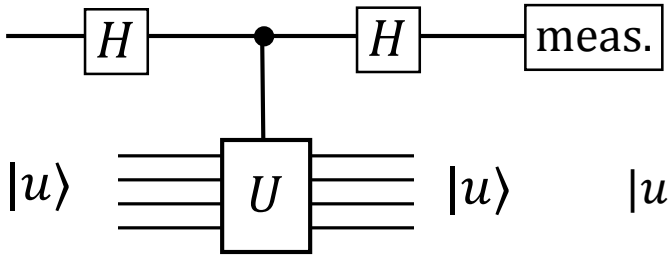
Consider a toy problem, which can be used in serious problems (period finding, etc.)



Suppose we know an eigenstate $|u\rangle$, but do not know the corresponding eigenvalue $e^{2\pi i\varphi}$
(since U is unitary, absolute value of eigenvalue is 1)

Goal: find φ

First idea:



$|u\rangle$ does not change, since eigenstate

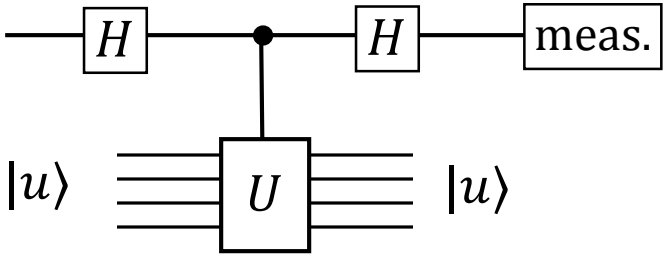
$$\frac{|0\rangle+|1\rangle}{\sqrt{2}}|u\rangle \xrightarrow{c\text{-}U} \frac{|0\rangle+|1\rangle\, e^{2\pi i\varphi}}{\sqrt{2}}|u\rangle \xrightarrow{H} \frac{(|0\rangle+|1\rangle)+(|0\rangle-|1\rangle)\, e^{2\pi i\varphi}}{2}|u\rangle =$$

$$= \left(|0\rangle\frac{1+e^{2\pi i\varphi}}{2}+|1\rangle\frac{1-e^{2\pi i\varphi}}{2}\right)|u\rangle$$

Measure many times, find probabilities $p(0)$ and $p(1)$

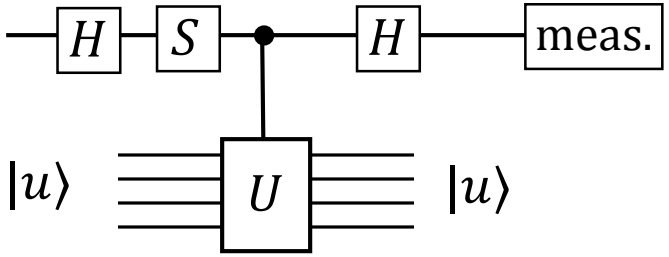
$$p(0)-p(1)=\cos(2\pi\varphi)$$

Phase estimation algorithm (cont.)



$$p(0) - p(1) = \cos(2\pi\varphi)$$

Now add *S*-gate $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$



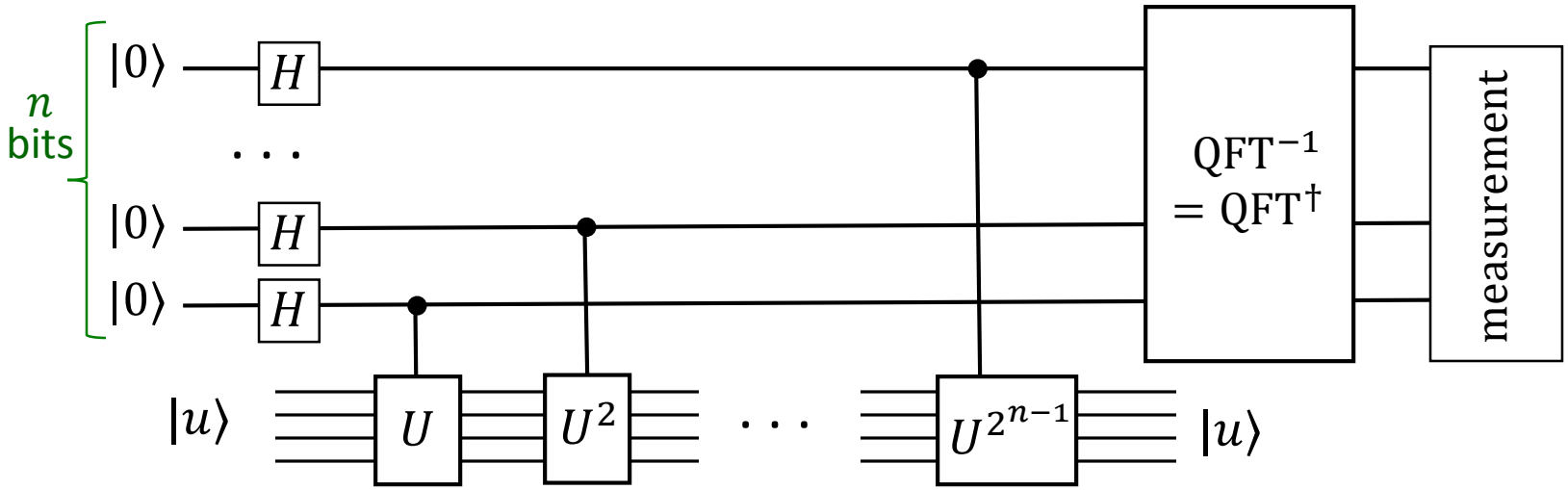
$$p(0) - p(1) = -\sin(2\pi\varphi)$$

Measuring many times, we can find φ accurately, but this is not fast (to find n bits of φ, we need $\sim 2^{2n}$ measurements)

Main idea: use c-U^2, c-U^4, c-U^8, etc. to find φ bit-by-bit (Kitaev)

Even better to use (inverse) QFT after that

Phase estimation algorithm (cont.)



State of the input register after c-U^k gates:

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i 2^{n-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{n-2}\varphi}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i\varphi}|1\rangle\right) =$$

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

This is just Fourier transform of $2^n\varphi$

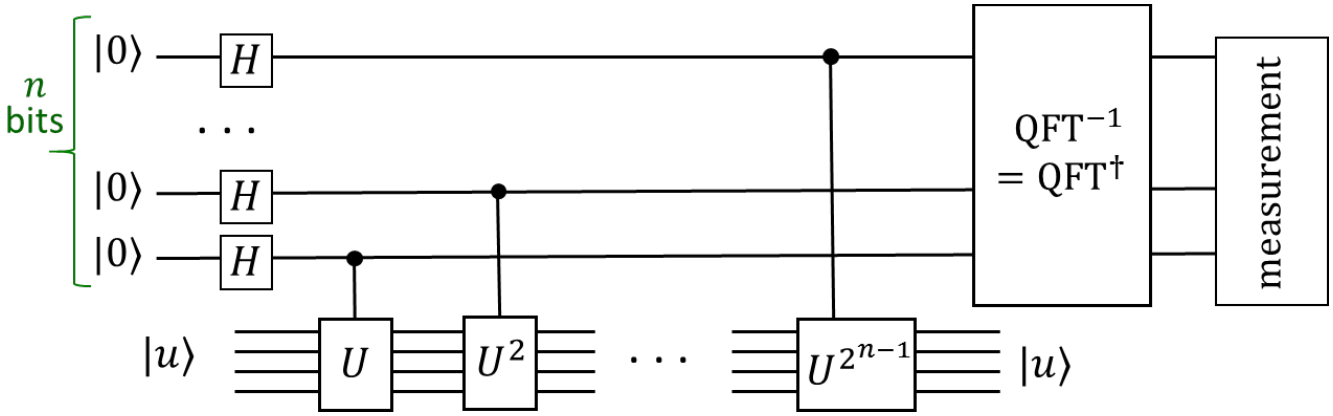
So, apply inverse QFT to get $2^n\varphi$

Exact result if φ has n-bit representation $0.\varphi_{n-1}\varphi_{n-2}\ldots\varphi_0$ (lower … upper)

If $2^n\varphi$ is not integer, then some errors.

Result: to find m bits of φ with probability $1-\varepsilon$, we need $n = m + \log(2+\frac{1}{2\varepsilon})$ qubits

Phase estimation algorithm (cont.)



Relation to period finding $f(x) = a^x \pmod N$

Define U as multiplication by a (mod N): $U|y\rangle = |ay \pmod N\rangle$. (unitary because a is coprime with N)

Then $U^r = \hat{1}$ for the period r, which we want to find.

Therefore eigenvalues of U are $e^{2\pi i\, j/r}$ for integer j.

So, finding the phase, we learn j/r (as in Shor's algorithm)

Therefore, phase estimation algorithms can be used for factoring integers.

It seems that for this algorithm we need to prepare an eigenstate $|u\rangle$. However, any state is a linear combination of eigenstates, so it does not matter (the algorithm will randomly find one of eigenstates of U). Natural to start with $|1\rangle$ (we need to avoid $|0\rangle$).

If output register starts with $|00..01\rangle$, then after c-U^k gates: $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle$