CS 258: Quantum Cryptography

Mark Zhandry

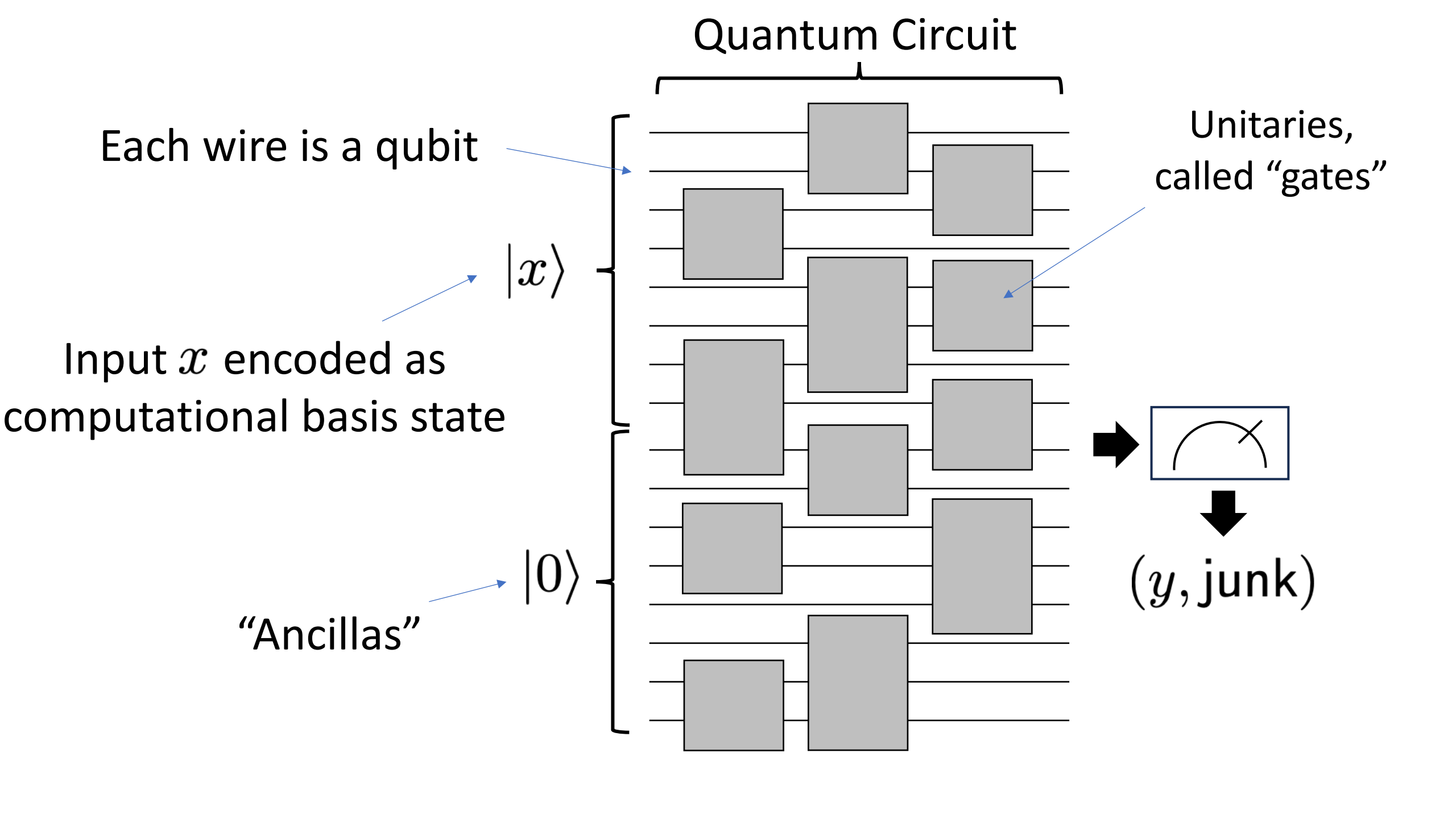
Previously…

In a nutshell, quantum computing is about using **interference** so that different computational paths leading to the correct answer constructively interfere, and computational paths leading to the wrong answers destructively interfere. The result is that the right answer is achieved with higher probability than what is possible classically

Obtaining the right answer with higher probability often means being able to obtain the right answer *faster*

Qubit

A qubit is just a 2-dimensional quantum system over $|0\rangle$ and $|1\rangle$

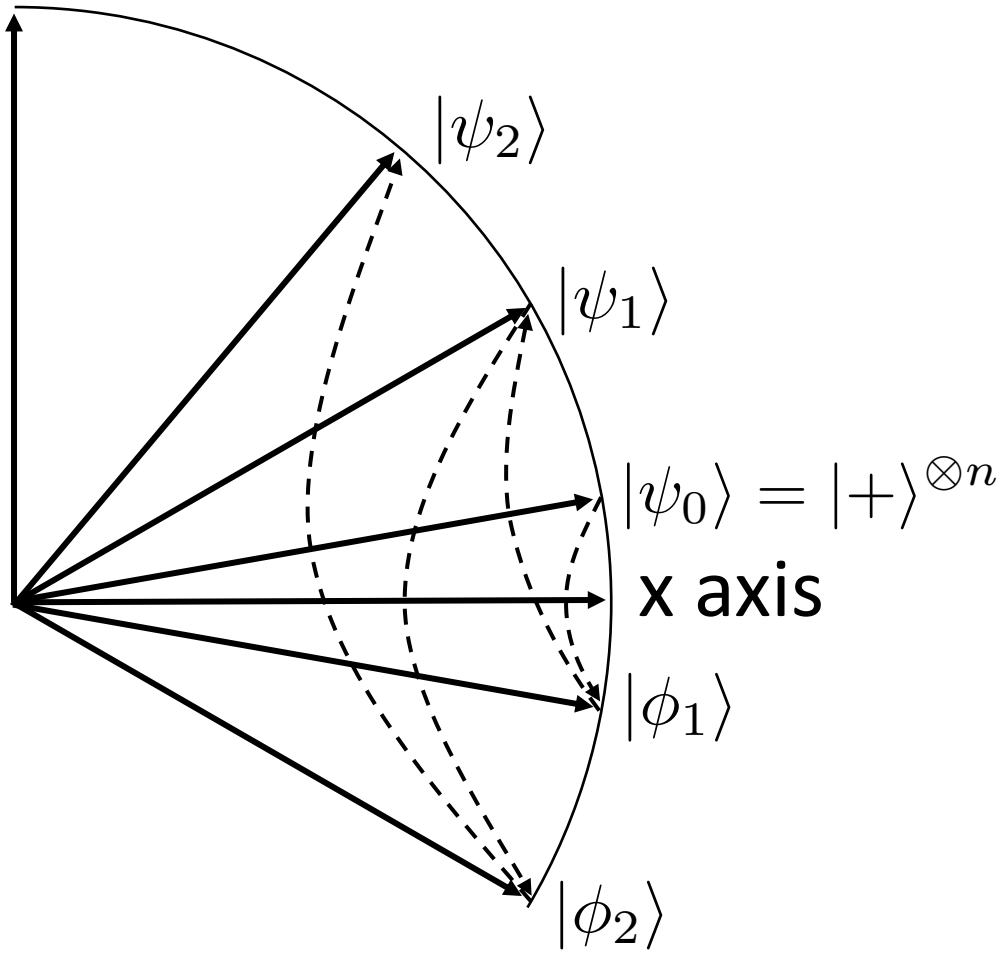

Quantum Circuit
Each wire is a qubit
Unitaries, called “gates”
$|x\rangle$
Input x encoded as computational basis state
$|0\rangle$
“Ancillas”
(y, junk)

Quantum >= Classical

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

Thm: There exists a quantum algorithm that performs $O(\sqrt{2^n})$ evaluations of U_f, and finds an x such that $f(x) = 1$ with probability $1 - O(2^{-n})$

y axis = $|x^*\rangle$



P_f reflects about x-axis

$\mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n}$ reflects about $|+\rangle^{\otimes n}$

Today: Quantum Period Finding (Shor’s Algorithm)

Period Finding

aka hidden subgroup problem

Let $\mathbb{G}$ be a discrete abelian group, written additively. e.g.

$$\mathbb{Z}^2 \qquad \mathbb{Z}_3^{17} \qquad \mathbb{Z}_{1041}$$

Def: A function $f : \mathbb{G} \to \mathcal{X}$ is **periodic** if there exists a $g \in \mathbb{G} \setminus \{0\}$ such that $f(x+g) = f(x)$ for all $x \in \mathbb{G}$. In this case, g is a **period** of f

Claim: periods of f form a subgroup

Proof: If g, h are periods, then

$$f(x+(g+h)) = f((x+g)+h) = f(x+g) = f(x)$$

$\Rightarrow$ $g+h$ is a period

Period Finding

aka hidden subgroup problem

Let $\mathbb{G}$ be a discrete abelian group, written additively. e.g.

$$\mathbb{Z}^2 \qquad \mathbb{Z}_3^{17} \qquad \mathbb{Z}_{1041}$$

Def: A function $f : \mathbb{G} \to \mathcal{X}$ is **periodic** if there exists a nonempty subgroup $\mathbb{H} \subset \mathbb{G}$ such that, for all $x \in \mathbb{G}$, $g \in \mathbb{H}$, $f(x+g) = f(x)$

Def: a periodic function $f : \mathbb{G} \to \mathcal{X}$ is **injective on its period** if, for all x, x' such that $x' - x \notin \mathbb{H}$

$$f(x) \neq f(x')$$

Period Finding

aka hidden subgroup problem

The period-finding problem (aka hidden subgroup problem) is to, given a periodic function f, find the group $\mathbb{H}$ of all periods

Technically, $\mathbb{H}$ is infinite, but we will ask instead for a minimal set of generators.

In the case of $\mathbb{Z}^n$, bit-length of generators may be unbounded, so some subtleties in how complexity is defined. But we won't need to worry about these

Classical Period Finding

Claim: Any black-box classical algorithm for period-finding requires exponentially-many queries

Proof idea: Consider a function $f : \mathbb{Z}_2^n \to \{0,1\}^m$ whose period is a random subgroup of order 2, $\mathbb{H} = \{0^n, x\}$

f is otherwise random on its period

For any classical query algorithm, there will be multiple possible choices consistent with query responses until it queries on y, z with $y \oplus z = x$. Exponentially unlikely until almost 2^n queries

Quantum Period Finding

Thm: There exists a QPT algorithm making only a polynomial number of queries which solves the period-finding problem with overwhelming probability

Warmup: the case of $\mathbb{Z}_2^n$

Also known as Simon's algorithm

Suppose $f : \mathbb{Z}_2^n \to \{0,1\}^m$ has period $\mathbb{H} = \{0^n, x\}$ and is injective on its period

Simon's Algorithm: Repeat the following several times:

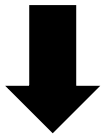
- Prepare $|+\rangle_{\mathcal{A}}^{\otimes n}|0^m\rangle_{\mathcal{B}} = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} |y\rangle_{\mathcal{A}}|0^m\rangle_{\mathcal{B}}$

- Apply U_f to obtain $\frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} |y\rangle_{\mathcal{A}}|f(y)\rangle_{\mathcal{B}}$

- Measure $\mathcal{B}$ → Measurement outcome z
  State collapses to $\left(\frac{1}{\sqrt{2}}|y\rangle + \frac{1}{\sqrt{2}}|y\oplus x\rangle\right)|z\rangle \qquad f(y) = f(y\oplus x) = z$

- Discard $|z\rangle$, apply $\mathbf{H}^{\otimes n}$, and measure → w

$$\mathbf{H}|b\rangle = \frac{1}{\sqrt{2}} \sum_{b'} (-1)^{bb'} |b'\rangle$$



$$\begin{aligned}
\mathbf{H}^{\otimes n}|y\rangle &= \left(\frac{1}{\sqrt{2}} \sum_{w_1} (-1)^{y_1 w_1} |w_1\rangle \right) \cdots \left(\frac{1}{\sqrt{2}} \sum_{w_n} (-1)^{y_n w_n} |w_n\rangle \right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{w} (-1)^{y_1 w_1 + \cdots + y_n w_n} |w\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{w} (-1)^{y \cdot w} |w\rangle
\end{aligned}$$

$$\mathbf{H}^{\otimes n}\left(\frac{1}{\sqrt{2}}|y\rangle+\frac{1}{\sqrt{2}}|y\oplus x\rangle\right)=\frac{1}{\sqrt{2^{n+1}}}\sum_{w}\left((-1)^{y\cdot w}+(-1)^{(y+x)\cdot w}\right)|w\rangle$$

$$=\frac{1}{\sqrt{2^{n+1}}}\sum_{w}(-1)^{y\cdot w}\left(1+(-1)^{x\cdot w}\right)|w\rangle$$

$w\cdot x=0 \bmod 2$ ➡ constructive interference

Weight on w: $\frac{2}{\sqrt{2^{n+1}}}=\frac{1}{\sqrt{2^{n-1}}}$

$w\cdot x=1 \bmod 2$ ➡ destructive interference

Weight on w: 0

$$\mathbf{H}^{\otimes n}\left(\frac{1}{\sqrt{2}}|y\rangle+\frac{1}{\sqrt{2}}|y\oplus x\rangle\right)=\frac{1}{\sqrt{2^{n+1}}}\sum_{w}\left((-1)^{y\cdot w}+(-1)^{(y+x)\cdot w}\right)|w\rangle$$

$$=\frac{1}{\sqrt{2^{n+1}}}\sum_{w}(-1)^{y\cdot w}\left(1+(-1)^{x\cdot w}\right)|w\rangle$$

$$=\frac{1}{\sqrt{2^{n-1}}}\sum_{w:w\cdot x=0 \bmod 2}(-1)^{y\cdot w}|w\rangle$$

Measure: obtain a random w that is orthogonal to x (mod 2)

Simon's Algorithm: generate $O(n)$ random w that are orthogonal to x (mod 2)

Assemble into matrix

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}$$

With high probability, the kernel of X will be exactly $\mathbb{H} = \{0^n, x\}$

Can find with Gaussian elimination (mod 2)

Suppose $f : \mathbb{Z}_2^n \to \{0,1\}^m$ has period a general $\mathbb{H}$ and is injective on its period

Algorithm still the same, kernel of $X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \end{pmatrix}$ is still $\mathbb{H}$

Shor's Algorithm: the case of general $f : \mathbb{G} \to \mathcal{X}$

Main tool: the Quantum Fourier Transform (QFT)

$$\mathsf{QFT}_q|x\rangle = \frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} e^{i2\pi xy/q}|y\rangle$$

$$\mathsf{QFT}_q = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1 & e^{i2\pi 1/q} & e^{i2\pi 2/q} & e^{i2\pi 3/q} & \cdots \\ 1 & e^{i2\pi 2/q} & e^{i2\pi 4/q} & e^{i2\pi 6/q} & \cdots \\ 1 & e^{i2\pi 3/q} & e^{i2\pi 6/q} & e^{i2\pi 9/q} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Main tool: the Quantum Fourier Transform (QFT)

For $x \in \mathbb{Z}_q^n$

$$\mathsf{QFT}_q^{\otimes n}|x\rangle = \frac{1}{\sqrt{q^n}} \sum_{y \in \mathbb{Z}_q^n} e^{i2\pi x \cdot y/q} |y\rangle$$

Observe: $\mathsf{QFT}_2 = \mathbf{H}$

QFT is unitary

$$\begin{aligned}
\mathsf{QFT}_q^\dagger \mathsf{QFT}_q |x\rangle &= \mathsf{QFT}_q^\dagger \left(\frac{1}{\sqrt{q}} \sum_y e^{i2\pi xy/q} |y\rangle \right) \\
&= \frac{1}{q} \sum_y \sum_z e^{i2\pi xy/q} e^{-i2\pi yz/q} |z\rangle \\
&= \frac{1}{q} \sum_z |z\rangle \left(\sum_y e^{i2\pi y(x-z)/q} \right)
\end{aligned}$$

QFT is unitary

Suppose $x = z \bmod q$ ➡ $x - z = kq$

$$\sum_y e^{i2\pi y(x-z)/q} = \sum_y e^{i2\pi ykq/q} = \sum_y e^{i2\pi yk} = \sum_y 1 = q$$

QFT is unitary

Suppose $x \neq z \bmod q$ ➡ $e^{i2\pi(x-z)/q} \neq 0$

$$\sum_y e^{i2\pi y(x-z)/q} = \sum_y \left(e^{i2\pi(x-z)/q}\right)^y = \frac{\left(e^{i2\pi(x-z)/q}\right)^q - 1}{\left(e^{i2\pi(x-z)/q}\right) - 1} = 0$$

QFT is unitary

$$
\begin{aligned}
\mathsf{QFT}_q^\dagger \mathsf{QFT}_q |x\rangle &= \mathsf{QFT}_q^\dagger \left(\frac{1}{\sqrt{q}} \sum_y e^{i2\pi xy/q} |y\rangle \right) \\
&= \frac{1}{q} \sum_y \sum_z e^{i2\pi xy/q} e^{-i2\pi yz/q} |z\rangle \\
&= \frac{1}{q} \sum_z |z\rangle \left(\sum_y e^{i2\pi y(x-z)/q} \right) \\
&= |x\rangle
\end{aligned}
$$

QFT and Coset States

Consider the group $\mathbb{Z}_q^n$

A coset state is a state of the form $\frac{1}{\sqrt{|\mathbb{H}|}} \sum_{g \in \mathbb{H}} |x + g \bmod q\rangle$

where $\mathbb{H}$ is a subgroup of $\mathbb{Z}_q^n$

$$\mathsf{QFT}_q^{\otimes n} \frac{1}{\sqrt{|\mathbb{H}|}} \sum_{g \in \mathbb{H}} |x + g \bmod q\rangle = \frac{1}{\sqrt{|\mathbb{H}| q^n}} \sum_y |y\rangle \left(\sum_{g \in \mathbb{H}} e^{i2\pi (x+g)\cdot y/q} \right)$$

$$= \frac{1}{\sqrt{|\mathbb{H}| q^n}} \sum_y |y\rangle e^{i2\pi x \cdot y/q} \left(\sum_{g \in \mathbb{H}} e^{i2\pi g \cdot y/q} \right)$$

Quotient group:

$$\mathbb{Z}_q^n/\mathbb{H} = \{y \in \mathbb{Z}_q^n : \forall g \in \mathbb{H}, y \cdot g = 0 \bmod q\}$$

$$|\mathbb{Z}_q^n/\mathbb{H}| = q^n/|\mathbb{H}|$$

Evaluating $\sum_{g \in \mathbb{H}} e^{i2\pi g \cdot y/q}$

For $y \in \mathbb{Z}_q^n / \mathbb{H}$, sum is $|\mathbb{H}|$

For $y \notin \mathbb{Z}_q^n / \mathbb{H}$, sum is 0

$$\mathsf{QFT}_q^{\otimes n} \frac{1}{\sqrt{|\mathbb{H}|}} \sum_{g \in \mathbb{H}} |x + g \bmod q\rangle = \frac{1}{\sqrt{|\mathbb{H}|q^n}} \sum_y |y\rangle \left(\sum_{g \in \mathbb{H}} e^{i2\pi(x+g)\cdot y/q} \right)$$

$$= \frac{1}{\sqrt{|\mathbb{H}|q^n}} \sum_y |y\rangle e^{i2\pi x \cdot y/q} \left(\sum_{g \in \mathbb{H}} e^{i2\pi g \cdot y/q} \right)$$

$$= \sqrt{\frac{|\mathbb{H}|}{q^n}} \sum_{y \in \mathbb{Z}_q^n / \mathbb{H}} |y\rangle e^{i2\pi x \cdot y/q}$$

Shor's Algorithm: Repeat the following several times:

- Prepare $\frac{1}{\sqrt{q^n}} \sum_{y \in \mathbb{Z}_q^n} |y\rangle_{\mathcal{A}} |0^m\rangle_{\mathcal{B}}$
- Apply U_f to obtain $\frac{1}{\sqrt{q^n}} \sum_{y} |y\rangle_{\mathcal{A}} |f(y)\rangle_{\mathcal{B}}$
- Measure $\mathcal{B}$ → Measurement outcome z
  State collapses to $\frac{1}{\sqrt{|\mathbb{H}|}} \sum_{g \in \mathbb{H}} |y + g \bmod q\rangle |z\rangle \qquad f(y) = z$
- Discard $|z\rangle$, apply $\mathsf{QFT}_q^{\otimes n}$, and measure → $w \in \mathbb{Z}_q^n / \mathbb{H}$

After polynomially-many steps, $\{w\}$ will generate $\mathbb{Z}_q^n/\mathbb{H}$

Can use Gaussian elimination to learn $\mathbb{H}$

We've only sketched the algorithm for $\mathbb{Z}_q^n$, but can be extended to any discrete abelian group

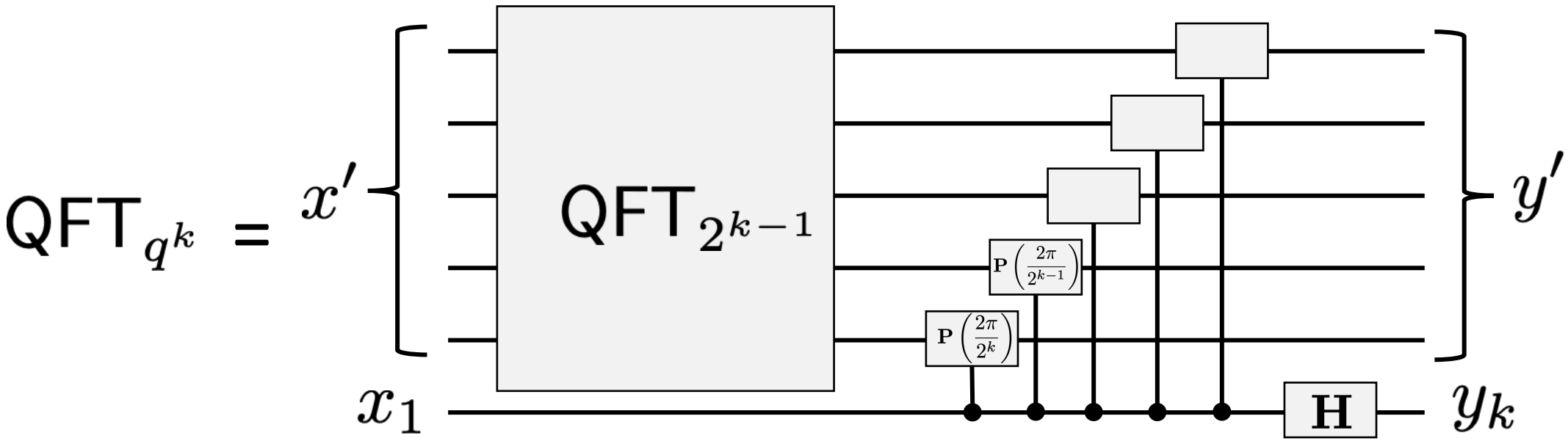
How to implement the general QFT

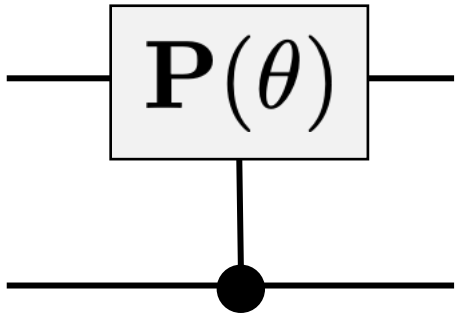
We will only consider the case $q = 2^k$

Recursive construction based on classical fast Fourier transforms

$$\mathsf{QFT}_{2^k}|2x'+x_1\rangle = \sum_{y_k,y'} |2^{k-1}y_k + y'\rangle e^{i2\pi(2x'+x_1)(2^{k-1}y_k+y')/2^k}$$

$$= \sum_{y_k,y'} |2^{k-1}y_k + y'\rangle \underbrace{\left(e^{i2\pi x'y'/2^{k-1}}\right.}_{\mathsf{QFT}_{2^{k-1}}|x'\rangle} \underbrace{e^{i2\pi x_1 y_k/2}}_{\mathbf{H}|x_1\rangle} \underbrace{\left.e^{i2\pi x_1 y'/2^k}\right)}_{\text{Phase correction}}$$





Control wire

Apply $\mathbf{P}(\theta)$ if control wire is 1

Application 1: Discrete Log

Recall Discrete Log Problem

Cyclic group $\mathbb{G}$ with generator g $(\mathbb{G} = \{1, g, g^2, g^3, \cdots\})$

Order of group: q

Discrete log problem: given $h = g^a$, find $a \in \mathbb{Z}_q$

Examples where Dlog assumed hard:

$\mathbb{Z}_p^*$ Elliptic curves over finite fields

Reducing Dlog to Period-Finding

Given $h = g^a$, define $f : \mathbb{Z}_q^2 \to \mathbb{G}$ $\qquad f(x, y) = g^x h^y$

Observe: $f(\ (x, y) + (ra, -r)\ ) = g^{x+ra} h^{y-r} = g^{x+ra} g^{ay-ar}$

$$= g^{x+ay} = g^x h^y = f(x, y)$$

Therefore, the period of f is $\mathbb{H} = \{(ra, -r)\}_{r \in \mathbb{Z}_q}$

Plus, f is injective on its period since g is a generator

Description of $\mathbb{H}$ reveals a

Application 2: Factoring

Factoring as Period Finding

Suppose for simplicity that $N = p_0 \times p_1$ for primes p_0, p_1

Choose a random $a \in \mathbb{Z}_N \setminus \{0\}$

Can assume $\mathsf{GCD}(a, N) = 1$, else $\mathsf{GCD}(a, N) \in \{p_0, p_1\}$

The set of such a is called $\mathbb{Z}_N^*$

Factoring as Period Finding

Suppose for simplicity that $N = p_0 \times p_1$ for primes p_0, p_1

Choose a random $a \in \mathbb{Z}_N^*$

Let $f_a : \mathbb{Z} \to \mathbb{Z}_N^*$ be $f_a(x) = a^x \bmod N$

Period finding on f_a reveals period of a , the smallest positive integer such that $a^x \bmod N = 1$

Observations

Let x_b be the period of a in $\mathbb{Z}_{p_b}^*$

Then $x = \mathsf{LCM}(x_0, x_1)$

$\mathbb{Z}_{p_b}^*$ is cyclic of order $p_b - 1$

➡ Write $a \bmod p_b$ as $g_b^{s_b} \bmod p_b$

where g_b is a generator of $\mathbb{Z}_{p_b}^*$

s_b is random in $\{0, 1, 2, \cdots p_b - 2\}$

Observations

y , odd

Write $x_b = 2^{k_b} r_b$ for odd b ➡ $x = 2^{\max(k_0,k_1)} \mathrm{LCM}(r_0, r_1)$

Claim: With probability at least $1/2$, $k_0 \neq k_1$

Proof: k_b is #{2's in $p_b - 1$} - #{2's in s_b }
Probability of any particular k_b is at most 1/2

Observations

Write $x_b = 2^{k_b} r_b$ for odd b ➡ $x = 2^{\max(k_0,k_1)} \underbrace{\mathrm{LCM}(r_0, r_1)}_{y\text{ , odd}}$

Assume $k_0 \neq k_1$, take $k_0 < k_1$

Observe: $x^{(2^{k_0} y)} \bmod p_0 = 1$

$x^{(2^{k_0} y)} \bmod p_1 \neq 1$

➡ $\mathrm{GCD}(x^{(2^{k_0} y)} \bmod N - 1, N) = p_0$

The Algorithm

Choose a random $a \in \mathbb{Z}_N^*$

Let $f_a : \mathbb{Z} \to \mathbb{Z}_N^*$ be $f_a(x) = a^x \bmod N$

Period finding on f_a reveals period x of a

For $i = 0, 1, 2, \cdots$ until $x/2^i$ is not an integer:

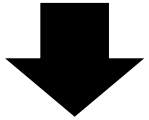
Compute $p = \mathsf{GCD}(a^{x/2^i} \bmod N - 1, N)$

If $p > 1$, output $\{p, N/p\}$

Keep trying new a until success

Impact on Cryptography

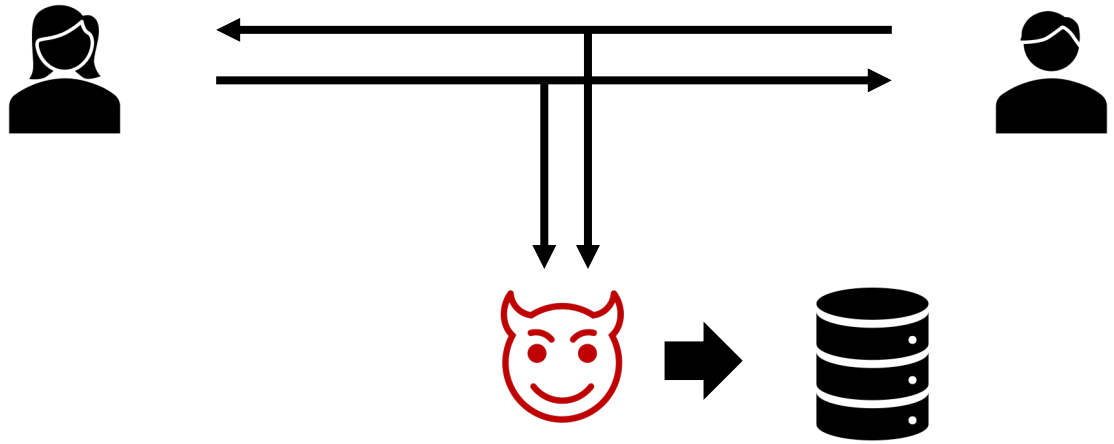
Unfortunately, all currently deployed public key cryptosystems rely on the hardness of either Discrete Log or Factoring



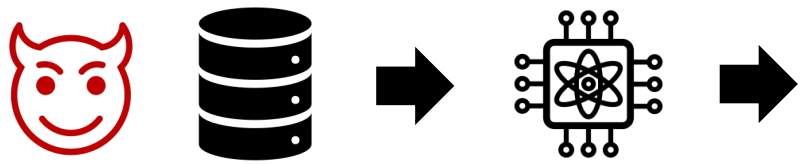
Basically all of our communication is broken once quantum computers are able to run Shor’s algorithm

Harvest-now-decrypt-later

Today:



Future:



Learn Alice and Bob’s private messages

If quantum computers will arrive in year T, and we have messages that we need to be secure for at least X years, must transition to post-quantum cryptosystems by year $T - X$

The situation is urgent, but we also can't rush since we don't want to accidentally introduce vulnerabilities when we do

A Cautionary Tale: The Case of SIKE

Supersingular Isogeny Key Exchange

2014: SIKE proposed, became one of the leading candidates for post-quantum key exchange

2022: SIKE completely broken by **classical** attack

If SIKE had been deployed in an effort to protect against quantum computers, we would've actually been compromised **now** against today's classical computers.

Starting Next Time: Candidate Post-Quantum Cryptosystems