How to prove you have built a quantum computer

Gregory D. Kahanamoku-Meyer

November 2, 2023

Introduction

… or, how did I get here?

Introduction

… or, how did I get here?

- Grew up and went to college in New England

Introduction

… or, how did I get here?

- Grew up and went to college in New England
- Recently completed PhD at UC Berkeley

Introduction

… or, how did I get here?

- Grew up and went to college in New England
- Recently completed PhD at UC Berkeley
- Spouse is a SOEST Early Career Research Fellow at UH Manoa

Introduction

... or, how did I get here?

- Grew up and went to college in New England
- Recently completed PhD at UC Berkeley
- Spouse is a SOEST Early Career Research Fellow at UH Manoa
- Currently living on O‘ahu while continuing my research remotely as a postdoc

Quantum computing: motivation

How hard is simulating quantum systems with (regular) computers?

Quantum computing: motivation

How hard is simulating quantum systems with (regular) computers?

Single quantum system with two states

Ex: spin-1/2 particle

Quantum computing: motivation

How hard is simulating quantum systems with (regular) computers?

Single quantum system with two states

Ex: spin-1/2 particle

Quantum state represented by
2 complex numbers

Quantum computing: motivation

How hard is simulating quantum systems with (regular) computers?

15 quantum particles with two states each

Ex: 15 spin-1/2 particles

Quantum computing: motivation

How hard is simulating quantum systems with (regular) computers?

15 quantum particles with two states each

Ex: 15 spin-1/2 particles

Quantum state represented by
$2^{15} \approx 30,000$ complex numbers

Quantum computing: motivation

How hard is simulating quantum systems with (regular) computers?

30 quantum particles with two states each

Ex: 30 spin-1/2 particles

Quantum computing: motivation

How hard is simulating quantum systems with (regular) computers?

30 quantum particles with two states each

Ex: 30 spin-1/2 particles

Quantum state represented by
$2^{30} \approx 1,000,000,000$ complex numbers

Quantum computing: motivation

Complexity grows exponentially with the number of particles!

Quantum computing: motivation

Complexity grows exponentially with the number of particles!

Can we use that complexity to perform computations?

Quantum computing: history

Early 90s: Theoretical algorithms for “bespoke” problems built for quantum computers

Quantum computing: history

Early 90s: Theoretical algorithms for "bespoke" problems built for quantum computers

Simon's problem

Given a function (implemented by a black box or oracle) $f : \{0,1\}^n \to \{0,1\}^n$ with the promise that, for some unknown $s \in \{0,1\}^n$, for all $x, y \in \{0,1\}^n$,

$$f(x) = f(y) \text{ if and only if } x \oplus y \in \{0^n, s\},$$

where $\oplus$ denotes bitwise XOR. The goal is to identify s by making as few queries to $f(x)$ as possible. Note that

$$a \oplus b = 0^n \text{ if and only if } a = b$$

Furthermore, for some x and s in $x \oplus y = s$, y is unique (not equal to x) if and only if $s \neq 0^n$. This means that f is two-to-one when $s \neq 0^n$, and one-to-one when $s = 0^n$. It is also the case that $x \oplus y = s$ implies $y = s \oplus x$, meaning that

$$f(x) = f(y) = f(x \oplus s)$$

Quantum computing: history

Mid 90s: Theoretical algorithms for real problems!

Quantum computing: history

Mid 90s: Theoretical algorithms for real problems!

Grover search

Faster searching

Shor's algorithm

Faster integer factorization

$$pq \rightarrow p \cdot q$$

Framing the question

Goal: construct a physical system that can actually run these algorithms!

Goal: construct a physical system that can actually run these algorithms!

Thursday, August 4, 2022
Today's Paper

The New York Times

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books

Theranos Leaves Biotech Business, Turns to Building Quantum Computers

- CEO Elizabeth Holmes states the emerging field of quantum computing will be a "new start" for the company
- Despite extensive fraud at previous company, investors inexplicably believe it's a good idea to dump millions of dollars into this new venture

This is not a real headline! It is a joke.

Framing the question

Goal: construct a physical system that can actually run these algorithms!

Suppose someone opens a cloud service to perform quantum computations.

Framing the question

Goal: construct a physical system that can actually run these algorithms!

Suppose someone opens a cloud service to perform quantum computations.

How do we test if they are really doing anything quantum?

With only classical questions and answers?

Framing the question

Goal: construct a physical system that can actually run these algorithms!

Suppose someone opens a cloud service to perform quantum computations.

How do we test if they are really doing anything quantum?

With only classical questions and answers?

Maybe we can use those algorithms I just mentioned?

Grover search

Cost to find the "good" value from N indices

Quantum

$\mathcal{O}(\sqrt{N})$ operations

Classical

$\mathcal{O}(N)$ operations

Grover search

Cost to find the “good” value from N indices

Quantum

$\mathcal{O}(\sqrt{N})$ operations

Classical

$\mathcal{O}(N)$ operations

Grover search

Cost to find the “good” value from N indices

Quantum

$\mathcal{O}(\sqrt{N})$ operations

Classical

$\mathcal{O}(N)$ operations

Grover search

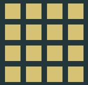
Cost to find the “good” value from N indices

Quantum

$\mathcal{O}(\sqrt{N})$ operations

Classical

$\mathcal{O}(N)$ operations



Grover search

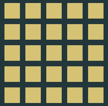
Cost to find the “good” value from N indices

Quantum

$\mathcal{O}(\sqrt{N})$ operations

Classical

$\mathcal{O}(N)$ operations



Grover search

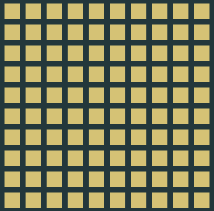
Cost to find the "good" value from N indices

Quantum

$\mathcal{O}(\sqrt{N})$ operations

Classical

$\mathcal{O}(N)$ operations



Challenge: Classical computers are fast!

Fewer quantum operations, but must account for differences in number of operations per second

Challenge: Classical computers are fast!

Fewer quantum operations, but must account for differences in number of operations per second

Quantum

$\mathcal{O}(\sqrt{N})$ operations

10^4 operations per second

Classical

$\mathcal{O}(N)$ operations

10^{10} operations per second

Challenge: Classical computers are fast!

Fewer quantum operations, but must account for differences in number of operations per second

Quantum

$\mathcal{O}(\sqrt{N})$ operations

10^4 operations per second

Classical

$\mathcal{O}(N)$ operations

10^{10} operations per second

Quantum speedups

Task	Theoretical speedup	Practical in 2023?
Grover search	Somewhat fewer ops.	Quantum computers too slow

Shor's algorithm

Goal: factor numbers $pq = p \cdot q$

Quantum

Classical

Shor's algorithm

Goal: factor numbers $pq = p \cdot q$

Quantum

Classical

Shor's algorithm

Goal: factor numbers $pq = p \cdot q$

Quantum

Classical

Shor's algorithm

Goal: factor numbers $pq = p \cdot q$

Quantum

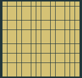
Classical

Shor's algorithm

Goal: factor numbers $pq = p \cdot q$

Quantum

Classical

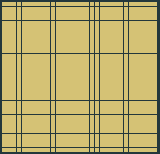


Shor's algorithm

Goal: factor numbers $pq = p \cdot q$

Quantum

Classical

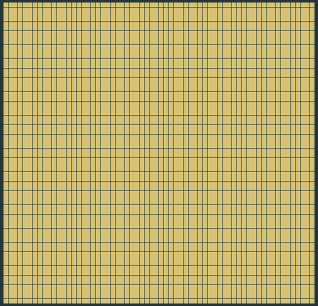


Shor's algorithm

Goal: factor numbers $pq = p \cdot q$

Quantum

Classical

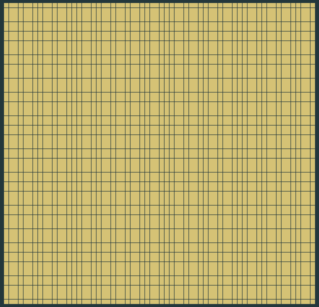


Challenge: quantum computers too noisy

Quantum information very fragile!

Quantum

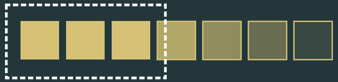
Classical



Challenge: quantum computers too noisy

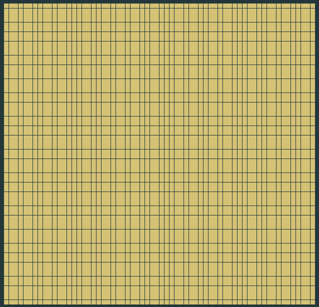
Quantum information very fragile! And devices are small!

Quantum



size of largest existing device

Classical



Quantum advantage in practice

Task	Theoretical speedup	Practical in 2023?
Grover search	Somewhat fewer ops.	Too slow, small and noisy
Shor’s factoring	Exponentially fewer ops.	Too small and noisy

Quantum advantage in practice

Task	Theoretical speedup	Practical in 2023?
Grover search	Somewhat fewer ops.	Too small, slow and noisy
Shor's factoring	Exponentially fewer ops.	Too small and noisy
Machine learning	Depends	Too small, slow and noisy
Chemistry	Depends	Too small, slow and noisy

Demonstrating “quantum advantage”

To prove we have built a quantum computer, the problem doesn’t have to be *useful*

Demonstrating “quantum advantage”

To prove we have built a quantum computer, the problem doesn’t have to be *useful*

> Is there *anything* current quantum computers can do that classical ones can’t?

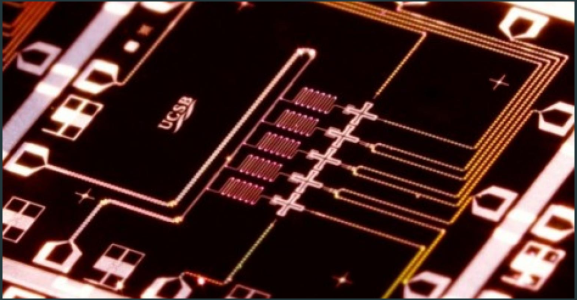
Demonstrating "quantum advantage"

To prove we have built a quantum computer, the problem doesn't have to be *useful*

Is there *anything* current quantum computers can do that classical ones can't?

10 years ago: nope!

Trivial to simulate!



Google/UCSB's 5-qubit chip

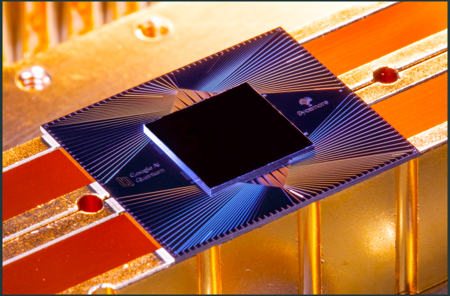
Demonstrating "quantum advantage"

To prove we have built a quantum computer, the problem doesn't have to be *useful*

Is there *anything* current quantum computers can do that classical ones can't?

Since 4 years ago: maybe??

Very hard to simulate!



Google's 53-qubit chip

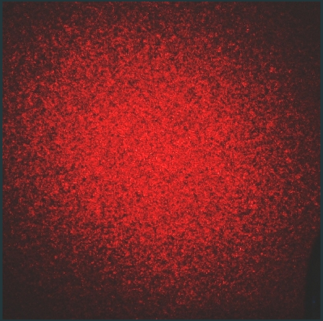
Random circuit sampling

What problem do we try to solve? Something quantum-related!

Random circuit sampling

What problem do we try to solve? Something quantum-related!

Sampling from an “speckle” (interference pattern)



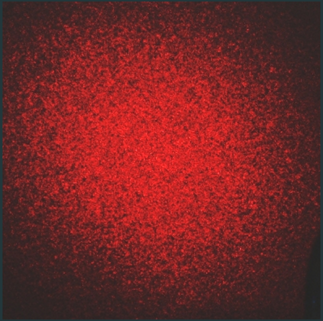
By Epzcaw - Own work, Public Domain,

https://commons.wikimedia.org/w/index.php?curid=4608610

Random circuit sampling

What problem do we try to solve? Something quantum-related!

Sampling from an "speckle" (interference pattern)



By Epzcaw - Own work, Public Domain,

https://commons.wikimedia.org/w/index.php?curid=4608610

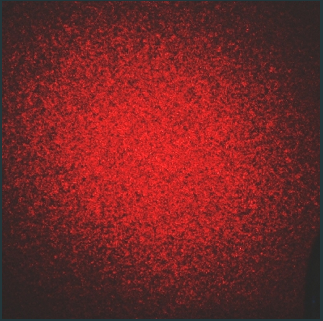
Mathematical problem:

1. Define some operations that generate a complicated quantum state

Random circuit sampling

What problem do we try to solve? Something quantum-related!

Sampling from an "speckle" (interference pattern)



By Epzcaw - Own work, Public Domain,

https://commons.wikimedia.org/w/index.php?curid=4608610

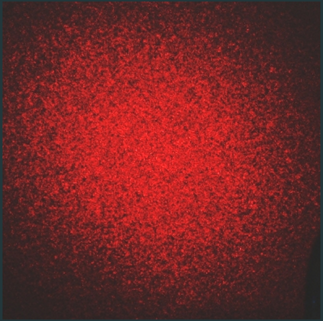
Mathematical problem:

1. Define some operations that generate a complicated quantum state
2. Quantum state defines a probability distribution of measurement outcomes

Random circuit sampling

What problem do we try to solve? Something quantum-related!

Sampling from an "speckle" (interference pattern)



By Epzcaw - Own work, Public Domain,

https://commons.wikimedia.org/w/index.php?curid=4608610

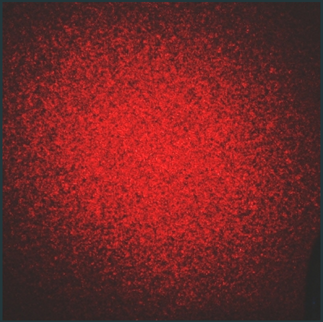
Mathematical problem:

1. Define some operations that generate a complicated quantum state
2. Quantum state defines a probability distribution of measurement outcomes
3. **Task:** Generate samples from that distribution

Random circuit sampling

What problem do we try to solve? Something quantum-related!

Sampling from an “speckle” (interference pattern)



By Epzcaw - Own work, Public Domain,

https://commons.wikimedia.org/w/index.php?curid=4608610

Mathematical problem:

1. Define some operations that generate a complicated quantum state
2. Quantum state defines a probability distribution of measurement outcomes
3. **Task:** Generate samples from that distribution

If distribution is complicated enough, generating samples is classically hard

Google’s results

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > articles > article

Article | Published: 23 October 2019

Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun

A subtle challenge: verification

Google published a bunch of samples from a “hard” probability distribution.

A subtle challenge: verification

Google published a bunch of samples from a “hard” probability distribution.

How do we confirm they actually came from that distribution?

A subtle challenge: verification

Google published a bunch of samples from a “hard” probability distribution.

How do we confirm they actually came from that distribution?

1. “Benchmark” quantum device by sampling from related but easy distributions

A subtle challenge: verification

Google published a bunch of samples from a “hard” probability distribution.

How do we confirm they actually came from that distribution?

1. “Benchmark” quantum device by sampling from related but easy distributions
2. Assume nothing weird happens when you switch to the hard distribution

Quantum advantage

Task	Theoretical speedup	Practical in 2023?
Grover search	Somewhat fewer ops.	Too small, slow and noisy
Shor's factoring	Exponentially fewer ops.	Too small and noisy
Machine learning	Depends	Too small, slow and noisy
Chemistry	Depends	Too small, slow and noisy
Random sampling	Exponentially fewer ops.	Yes, but can't check answer

Verifiable quantum advantage

We want a problem that is **hard to classically solve**, but **easy to classically check**

Verifiable quantum advantage

We want a problem that is **hard to classically solve**, but **easy to classically check**

Factoring and search are such problems!

Verifiable quantum advantage

We want a problem that is **hard to classically solve**, but **easy to classically check**

Factoring and search are such problems!

But we also want achievable on near-term quantum device

NISQ verifiable quantum advantage

NISQ = "noisy intermediate-scale quantum"

Sampling problems

e.g. random circuits, Boson sampling, ...

✓ NISQ feasible
✗ Efficiently verifiable

Number theory problems

e.g. factoring, discrete logarithm, ...

✗ NISQ feasible
✓ Efficiently verifiable

add structure

make less costly

???

✓ NISQ feasible
✓ Efficiently verifiable

Adding structure to sampling problems

Example: evolve a quantum system under "IQP" Hamiltonians (products of Pauli X's)

$$H = X_0X_1X_3 + X_1X_2X_4X_5 + \cdots \tag{1}$$

Adding structure to sampling problems

Example: evolve a quantum system under "IQP" Hamiltonians (products of Pauli X's)

$$H = X_0 X_1 X_3 + X_1 X_2 X_4 X_5 + \cdots \tag{1}$$

[Shepherd, Bremner 2008]: Hide a secret in H, such that evolving and sampling gives results correlated with secret

Adding structure to sampling problems

Example: evolve a quantum system under "IQP" Hamiltonians (products of Pauli X's)

$$H = X_0 X_1 X_3 + X_1 X_2 X_4 X_5 + \cdots \tag{1}$$

[Shepherd, Bremner 2008]: Hide a secret in H, such that evolving and sampling gives results correlated with secret

[Bremner, Josza, Shepherd 2010]: classically simulating IQP Hamiltonians is hard

Adding structure to sampling problems

Example: evolve a quantum system under "IQP" Hamiltonians (products of Pauli X's)

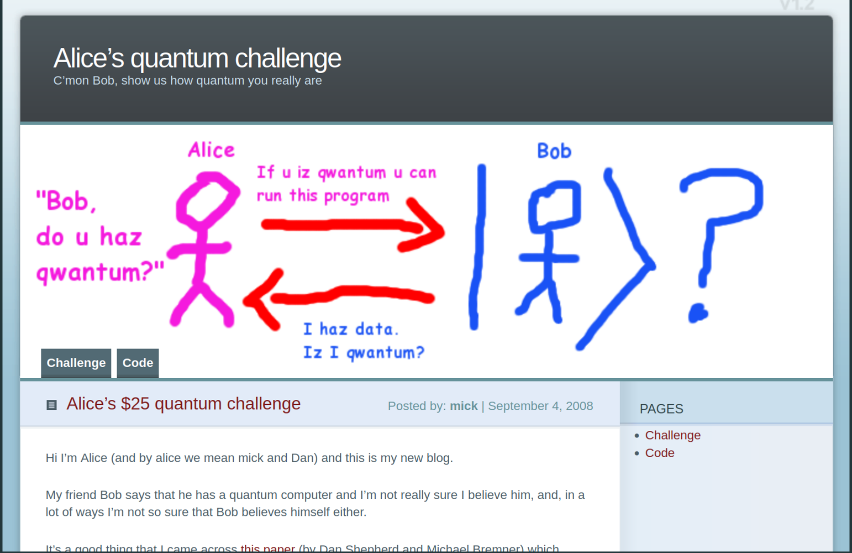
$$H = X_0 X_1 X_3 + X_1 X_2 X_4 X_5 + \cdots \tag{1}$$

[Shepherd, Bremner 2008]: Hide a secret in H, such that evolving and sampling gives results correlated with secret

[Bremner, Josza, Shepherd 2010]: classically simulating IQP Hamiltonians is hard

But how sure are we that the secret is really hidden?

The $25 challenge



Alice's quantum challenge

C'mon Bob, show us how quantum you really are

Challenge | Code

Alice's $25 quantum challenge

Posted by: **mick** | September 4, 2008

Hi I'm Alice (and by alice we mean mick and Dan) and this is my new blog.

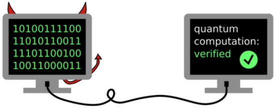
My friend Bob says that he has a quantum computer and I'm not really sure I believe him, and, in a lot of ways I'm not so sure that Bob believes himself either.

It's a good thing that I came across this paper (by Dan Shepherd and Michael Bremner) which

PAGES

- Challenge
- Code

Classical algorithm to extract secret



PAPER

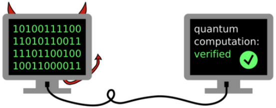
Forging quantum data: classically defeating an IQP-based quantum test

Gregory D. Kahanamoku-Meyer,

Quantum 7, 1107 (2023).

Recently, quantum computing experiments have for the first time exceeded the capability of classical computers to perform certain computations – a milestone termed "quantum computational adv...

Classical algorithm to extract secret



PAPER

Forging quantum data: classically defeating an IQP-based quantum test

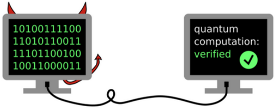
Gregory D. Kahanamoku-Meyer,

Quantum 7, 1107 (2023).

Recently, quantum computing experiments have for the first time exceeded the capability of classical computers to perform certain computations – a milestone termed "quantum computational adv...

Adding structure opens opportunities for classical cheating

Classical algorithm to extract secret



PAPER

Forging quantum data: classically defeating an IQP-based quantum test

Gregory D. Kahanamoku-Meyer,

Quantum 7, 1107 (2023).

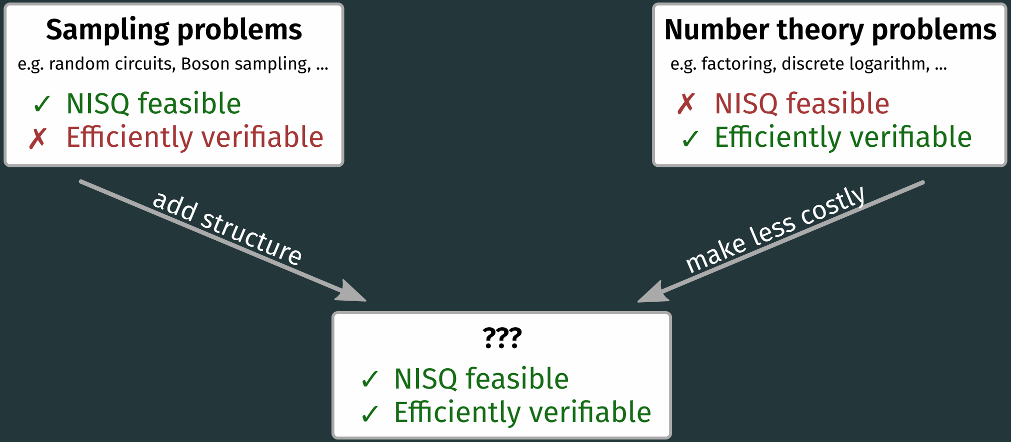
Recently, quantum computing experiments have for the first time exceeded the capability of classical computers to perform certain computations – a milestone termed "quantum computational adv...

Adding structure opens opportunities for classical cheating

[Bremner, Cheng, Ji 2023]: New scheme where the secret is (hopefully) hidden better

NISQ verifiable quantum advantage

NISQ = “noisy intermediate-scale quantum”



Making number theory problems less costly

Generating a quantum state that involves the factors is easy—
getting the factors out as classical values is the hard part!

Making number theory problems less costly

Generating a quantum state that involves the factors is easy—
getting the factors out as classical values is the hard part!

Idea from cryptography: zero-knowledge proof

Zero-knowledge proofs: differentiating colors

Challenge: Proving two balls are different colors

Zero-knowledge proofs: differentiating colors

Challenge: Proving two balls are different colors **without** actually telling them the colors?

Zero-knowledge proofs: differentiating colors

Challenge: Proving two balls are different colors **without** actually telling them the colors?

Solution:

1. They show you one ball, then hide it behind their back

Zero-knowledge proofs: differentiating colors

Challenge: Proving two balls are different colors **without** actually telling them the colors?

Solution:

1. They show you one ball, then hide it behind their back
2. They pull out another, you tell them same or different

Zero-knowledge proofs: differentiating colors

Challenge: Proving two balls are different colors **without** actually telling them the colors?

Solution:

1. They show you one ball, then hide it behind their back
2. They pull out another, you tell them same or different

This constitutes a **zero-knowledge interactive proof**.

Zero-knowledge proofs: differentiating colors

Challenge: Proving two balls are different colors
without actually telling them the colors?

Solution:

1. They show you one ball, then hide it behind their back
2. They pull out another, you tell them same or different

This constitutes a **zero-knowledge interactive proof**.

Seeing color $\Leftrightarrow$ Quantum capability

Zero-knowledge proofs: differentiating colors

Challenge: Proving two balls are different colors
without actually telling them the colors?

Solution:

1. They show you one ball, then hide it behind their back
2. They pull out another, you tell them same or different

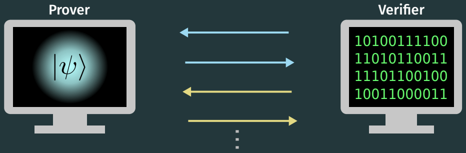
This constitutes a **zero-knowledge interactive proof**.

Seeing color $\Leftrightarrow$ Quantum capability

Goal: find protocol **as verifiable and classically hard as factoring**—
but **less expensive than actually finding factors (via Shor)**

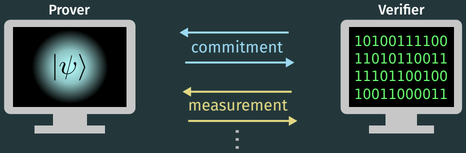
Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier



Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier

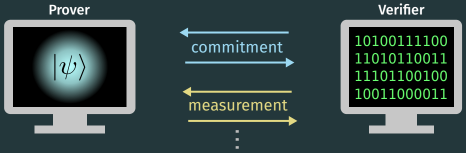


Round 1: Prover commits to holding a specific quantum state

Round 2: Verifier asks for measurement in random basis, prover performs it

Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier



Round 1: Prover commits to holding a specific quantum state

Round 2: Verifier asks for measurement in random basis, prover performs it

By randomizing choice of basis and repeating interaction,
can ensure prover actually has the promised quantum state

Brakerski, Christiano, Mahadev, Vidick, Vazirani '18 (arXiv:1804.00640).

Can be extended to verify arbitrary quantum computations! (arXiv:1804.01082)

Commitment: a secret quantum state

How does the prover commit to a state?

Consider a **2-to-1** function f:
for all y in range of f, there exist (x_0, x_1) such that $y = f(x_0) = f(x_1)$.

How does the prover commit to a state?

Consider a **2-to-1** function f:
for all y in range of f, there exist (x_0, x_1) such that $y = f(x_0) = f(x_1)$.

Generate entangled superposition $\sum_x |x\rangle |f(x)\rangle$ $\xleftarrow{\quad f \quad}$ Pick 2-to-1 function f

Commitment: a secret quantum state

How does the prover commit to a state?

Consider a **2-to-1** function f:
for all y in range of f, there exist (x_0, x_1) such that $y = f(x_0) = f(x_1)$.

Prover		Verifier
Generate entangled superposition $\sum_x \lvert x\rangle \lvert f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick 2-to-1 function f
Measure 2nd register as y	$\xrightarrow{\quad y \quad}$	Store y as commitment

Prover has committed to the state $(|x_0\rangle + |x_1\rangle)\,|y\rangle$

State commitment (round 1): trapdoor claw-free functions

Prover has committed to $(|x_0\rangle + |x_1\rangle) |y\rangle$ with $y = f(x_0) = f(x_1)$

State commitment (round 1): trapdoor claw-free functions

Prover has committed to $(|x_0\rangle + |x_1\rangle)\,|y\rangle$ with $y = f(x_0) = f(x_1)$

Source of power: cryptographic properties of 2-to-1 function f

State commitment (round 1): trapdoor claw-free functions

Prover has committed to $(|x_0\rangle + |x_1\rangle)\,|y\rangle$ with $y = f(x_0) = f(x_1)$

Source of power: cryptographic properties of 2-to-1 function f

- **"Claw-free"**: It is cryptographically hard to find any pair of colliding inputs

State commitment (round 1): trapdoor claw-free functions

Prover has committed to $(|x_0\rangle + |x_1\rangle)\,|y\rangle$ with $y = f(x_0) = f(x_1)$

Source of power: cryptographic properties of 2-to-1 function f

- **“Claw-free”**: It is cryptographically hard to find any pair of colliding inputs
- **Trapdoor**: With the secret key, easy to classically compute the two inputs mapping to any output

State commitment (round 1): trapdoor claw-free functions

Prover has committed to $(|x_0\rangle + |x_1\rangle)\,|y\rangle$ with $y = f(x_0) = f(x_1)$

Source of power: cryptographic properties of 2-to-1 function f

- **"Claw-free"**: It is cryptographically hard to find any pair of colliding inputs
- **Trapdoor**: With the secret key, easy to classically compute the two inputs mapping to any output

Cheating classical prover can't forge the state;
classical verifier can determine state using trapdoor.

State commitment (round 1): trapdoor claw-free functions

Prover has committed to $(|x_0\rangle + |x_1\rangle)\,|y\rangle$ with $y = f(x_0) = f(x_1)$

Source of power: cryptographic properties of 2-to-1 function f

- **"Claw-free"**: It is cryptographically hard to find any pair of colliding inputs
- **Trapdoor**: With the secret key, easy to classically compute the two inputs mapping to any output

Cheating classical prover can't forge the state;
classical verifier can determine state using trapdoor.

Generating a valid state without trapdoor uses superposition + wavefunction collapse—inherently quantum!

Trapdoor claw-free function example

$$f(x) = x^2 \bmod N, \text{ where } N = pq$$

Trapdoor claw-free function example

$$f(x) = x^2 \bmod N, \text{ where } N = pq$$

- **Claw-free:** Easy to compute p, q given a colliding pair—thus finding collisions is as hard as factoring

Trapdoor claw-free function example

$$f(x) = x^2 \bmod N, \text{ where } N = pq$$

- **Claw-free:** Easy to compute p, q given a colliding pair—thus finding collisions is as hard as factoring

Example: Let $p = 5$, $q = 7$; then $pq = 35$.

Trapdoor claw-free function example

$$f(x) = x^2 \bmod N, \text{ where } N = pq$$

- **Claw-free:** Easy to compute p, q given a colliding pair—thus finding collisions is as hard as factoring

Example: Let $p = 5$, $q = 7$; then $pq = 35$.
We have $4^2 \equiv 11^2 \equiv 16 \pmod{35}$; and $11 - 4 = 7$

How to implement it?

To generate the entangled superposition:

Need to square a number on a quantum computer!

How to implement it?

To generate the entangled superposition:

Need to square a number on a quantum computer!

Idea: use the same circuits that we do in classical computers?

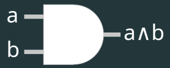
Challenge: reversibility

Coherent quantum circuits must be reversible!

Challenge: reversibility

Coherent quantum circuits must be reversible!

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

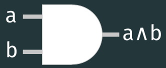


Classical AND

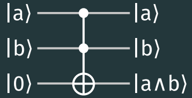
Challenge: reversibility

Coherent quantum circuits must be reversible!

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1



Classical AND

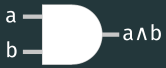


Quantum AND (Toffoli)

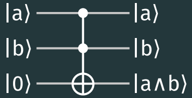
Challenge: reversibility

Coherent quantum circuits must be reversible!

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1



Classical AND



Quantum AND (Toffoli)

If you’re not careful, you will use up all of your precious qubits storing this “garbage data”!

Result: fast quantum multiplication with little space overhead

Result: fast quantum multiplication with little space overhead

Previous best:

Efficiently multiplying two 2048-bit numbers required over **12,000** extra qubits

Result: fast quantum multiplication with little space overhead

Previous best:

Efficiently multiplying two 2048-bit numbers required over **12,000** extra qubits

New paper (in prep.):

Efficiently multiplication with just **1 extra qubit**

Result: fast quantum multiplication with little space overhead

Previous best:

Efficiently multiplying two 2048-bit numbers required over **12,000** extra qubits

New paper (in prep.):

Efficiently multiplication with just **1 extra qubit**

Applications include proving "quantumness" but also factoring and other algorithms!

Putting it all together

Using Shor's factoring algorithm to prove you are quantum:

$\sim 10,000,000,000$ quantum operations

Putting it all together

Using Shor's factoring algorithm to prove you are quantum:

$\sim 10,000,000,000$ quantum operations

Using the new protocol:

$\sim 2,000,000$ quantum operations

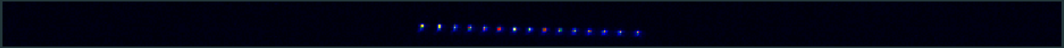
Proof-of-principle experiment

Trapped ions at the University of Maryland

Proof-of-principle experiment

Trapped ions at the University of Maryland

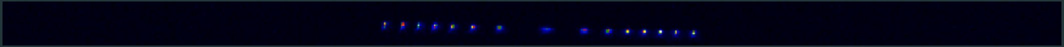
For interactive protocol, need to **measure a subset** of the quantum particles!



Proof-of-principle experiment

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!



Proof-of-principle experiment

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!

Proof-of-principle experiment

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!

Proof-of-principle experiment

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!

Proof-of-principle experiment

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!

Proof-of-principle experiment

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!

Proof-of-principle experiment

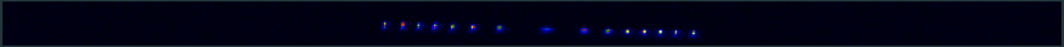
Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!

Proof-of-principle experiment

Trapped ions at the University of Maryland

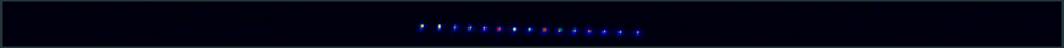
For interactive protocol, need to **measure a subset** of the quantum particles!



Proof-of-principle experiment

Trapped ions at the University of Maryland

For interactive protocol, need to **measure a subset** of the quantum particles!



Summary

- Current generation quantum computers: **slow, small, and noisy**

Summary

- Current generation quantum computers: **slow, small, and noisy**
- Large recent sampling experiments have **no way to check the answer**

Summary

- Current generation quantum computers: **slow, small, and noisy**
- Large recent sampling experiments have **no way to check the answer**
- Adding structure to those experiments can allow **classical spoofing**

Summary

- Current generation quantum computers: **slow, small, and noisy**
- Large recent sampling experiments have **no way to check the answer**
- Adding structure to those experiments can allow **classical spoofing**
- **Interactive protocols** give us “proofs of quantumness” that can be efficiently checked

Summary

- Current generation quantum computers: **slow, small, and noisy**
- Large recent sampling experiments have **no way to check the answer**
- Adding structure to those experiments can allow **classical spoofing**
- **Interactive protocols** give us “proofs of quantumness” that can be efficiently checked
- Using them requires doing **multiplication**—it can be performed efficiently with just 1 extra qubit

Summary

- Current generation quantum computers: **slow, small, and noisy**
- Large recent sampling experiments have **no way to check the answer**
- Adding structure to those experiments can allow **classical spoofing**
- **Interactive protocols** give us "proofs of quantumness" that can be efficiently checked
- Using them requires doing **multiplication**—it can be performed efficiently with just 1 extra qubit

Thank you!