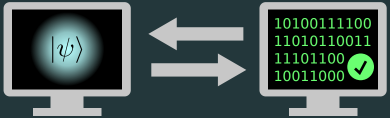
Classical verification of quantum computation



Greg Kahanamoku-Meyer

May 3, 2022

Focus of today

How can we demonstrate that a supposed "quantum computer" is actually doing something non-classical?

Focus of today

How can we demonstrate that a supposed "quantum computer" is actually doing something non-classical?

… or …

How can we demonstrate that *quantum computing in practice* can do something non-classical?

Focus of today

How can we demonstrate that a supposed "quantum computer" is actually doing something non-classical?

… or …

How can we demonstrate that *quantum computing in practice* can do something non-classical?

Setting:

- Single quantum "prover" (*computational* demonstration)

Focus of today

How can we demonstrate that a supposed "quantum computer" is actually doing something non-classical?

… or …

How can we demonstrate that *quantum computing in practice* can do something non-classical?

Setting:

- Single quantum "prover" (*computational* demonstration)
- "Verifier" + communication is entirely classical

Focus of today

How can we demonstrate that a supposed "quantum computer" is actually doing something non-classical?

... or ...

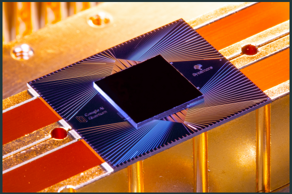
How can we demonstrate that *quantum computing in practice* can do something non-classical?

Setting:

- Single quantum "prover" (*computational* demonstration)
- "Verifier" + communication is entirely classical
- No assumptions about how prover works

Quantum computational advantage

Experiments claiming that their output can’t be simulated classically:



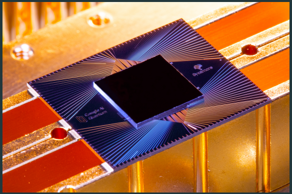
Random circuit sampling
[Google, 2019]

Gaussian boson sampling
[USTC, 2020]

...

Quantum computational advantage

Experiments claiming that their output can't be simulated classically:



Random circuit sampling
[Google, 2019]

Gaussian boson sampling
[USTC, 2020]

• • •

- How hard is it *really* to classically simulate?
- If indeed we can't simulate, how do we check that it's *correct*?

How hard is it to classically simulate?

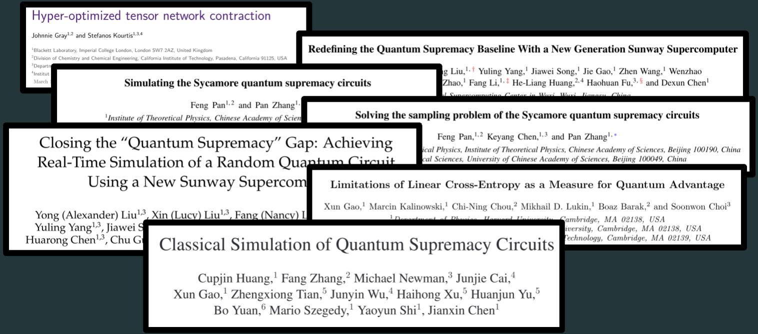
Focusing on Google's random circuit sampling experiment with 53 qubits:

Complexity theory suggests it's hard.

How hard is it to classically simulate?

Focusing on Google's random circuit sampling experiment with 53 qubits:

Complexity theory suggests it's hard. But...



What does it mean for a computation to be classically hard?

What does it mean to be classically hard?

Complexity theory

All about asymptotics. Example:

"Simulating the generic evolution of n qubits takes time that scales as $\mathcal{O}(2^n)$"

What does it mean to be classically hard?

Complexity theory

All about asymptotics. Example:

“Simulating the generic evolution of n qubits takes time that scales as $\mathcal{O}(2^n)$”

“hard” $\sim$ “superpolynomial”

What does it mean to be classically hard?

Complexity theory

All about asymptotics. Example:

"Simulating the generic evolution of n qubits takes time that scales as $\mathcal{O}(2^n)$"

"hard" $\sim$ "superpolynomial"

In practice

We care about actual resource costs for *a specific instance* of the problem. Ex:

"Simulating this depth-20 circuit on 20 qubits takes 10 minutes." (not hard)

What does it mean to be classically hard?

Complexity theory

All about asymptotics. Example:

"Simulating the generic evolution of n qubits takes time that scales as $\mathcal{O}(2^n)$"

"hard" $\sim$ "superpolynomial"

In practice

We care about actual resource costs for *a specific instance* of the problem. Ex:

"Simulating this depth-20 circuit on 20 qubits takes 10 minutes." (not hard)

"hard" $\sim$ "takes unrealistic resources"

What does it mean to be classically hard?

Complexity theory

All about asymptotics. Example:

"Simulating the generic evolution of n qubits takes time that scales as $\mathcal{O}(2^n)$"

"hard" $\sim$ "superpolynomial"

In practice

We care about actual resource costs for *a specific instance* of the problem. Ex:

"Simulating this depth-20 circuit on 20 qubits takes 10 minutes." (not hard)

"hard" $\sim$ "takes unrealistic resources"

Takeaway: Complexity theory tells us how the hardness of a problem *scales*, but not the actual cost for specific instances.

What does it mean to be classically hard?

Complexity theory

All about asymptotics. Example:

"Simulating the generic evolution of n qubits takes time that scales as $\mathcal{O}(2^n)$"

"hard" $\sim$ "superpolynomial"

In practice

We care about actual resource costs for *a specific instance* of the problem. Ex:

"Simulating this depth-20 circuit on 20 qubits takes 10 minutes." (not hard)

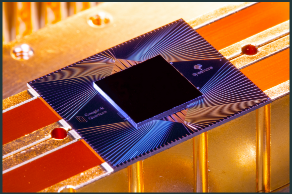
"hard" $\sim$ "takes unrealistic resources"

Takeaway: Complexity theory tells us how the hardness of a problem *scales*, but not the actual cost for specific instances.

Best strategy for finding cost in practice: **have a bunch of people try it.**

Quantum computational advantage

Experiments claiming that their output can't be simulated classically:



Random circuit sampling
[Google, 2019]

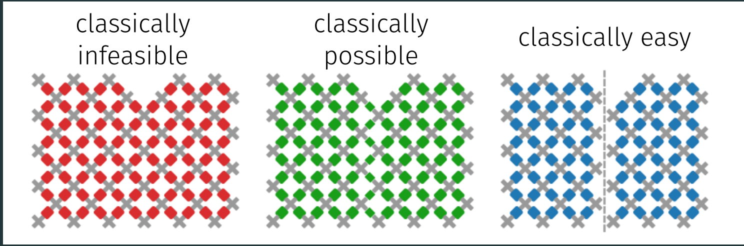
Gaussian boson sampling
[USTC, 2020]

…

- How hard is it *really* to classically simulate?
- If indeed we can't simulate, how do we check that it's *correct*?

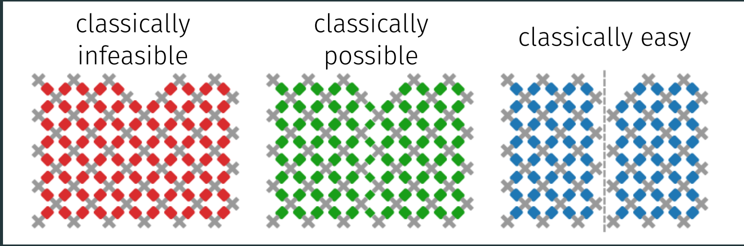
Random circuit sampling: checking correctness

Random circuit sampling: checking correctness



Idea: **extrapolate** correctness from simpler circuits.

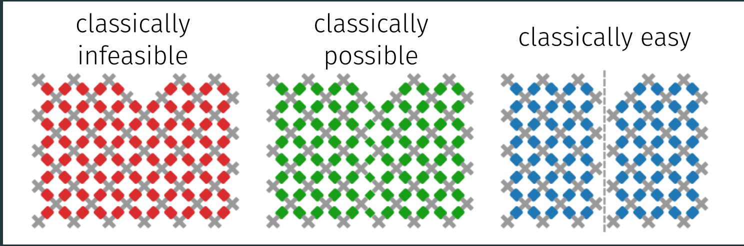
Random circuit sampling: checking correctness



Idea: **extrapolate** correctness from simpler circuits.

“The device works correctly on the easy ones, so it probably also works on the hard one”

Random circuit sampling: checking correctness



Idea: **extrapolate** correctness from simpler circuits.

"The device works correctly on the easy ones, so it probably also works on the hard one"

Ideally:

- Remove need for extrapolations/assumptions in verification process
- Not need a supercomputer to do it

Robust, verifiable quantum computational advantage

We want a test with three properties:

Robust, verifiable quantum computational advantage

We want a test with three properties:

- Easy for quantum device to pass

Robust, verifiable quantum computational advantage

We want a test with three properties:

- Easy for quantum device to pass
- Hard for classical computer to pass*

* with well-studied practical hardness!

Robust, verifiable quantum computational advantage

We want a test with three properties:

- Easy for quantum device to pass
- Hard for classical computer to pass*
- Easy for classical computer to verify

* with well-studied practical hardness!

Robust, verifiable quantum computational advantage

We want a test with three properties:

- **Easy** for near-term quantum device to pass
- **Hard** for classical supercomputer to pass*
- **Easy** for classical laptop computer to verify

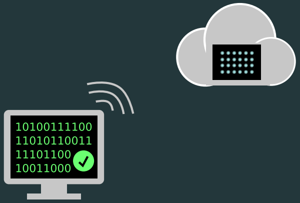
* with well-studied practical hardness!

Robust, verifiable quantum computational advantage

We want a test with three properties:

- **Easy** for near-term quantum device to pass
- **Hard** for classical supercomputer to pass*
- **Easy** for classical laptop computer to verify

* with well-studied practical hardness!



Remote: validate an untrusted quantum device over the internet

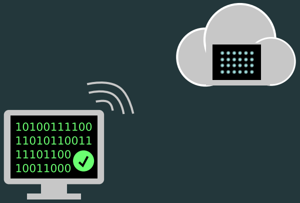
"Website proves its power to user"

Robust, verifiable quantum computational advantage

We want a test with three properties:

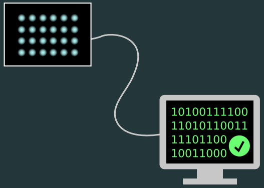
- **Easy** for near-term quantum device to pass
- **Hard** for classical supercomputer to pass*
- **Easy** for classical laptop computer to verify

* with well-studied practical hardness!



Remote: validate an untrusted quantum device over the internet

"Website proves its power to user"



Local: robust demonstration of the power of quantum computation

"Qubits prove their power to humanity"

Connection to cryptography

> "Making sure certain things computationally hard, while keeping others easy."

Connection to cryptography

> “Making sure certain things computationally hard, while keeping others easy.”

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Connection to cryptography

> "Making sure certain things computationally hard, while keeping others easy."

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Many other applications: authentication, digital signatures, multi-party computation, ...

Connection to cryptography

> "Making sure certain things computationally hard, while keeping others easy."

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Many other applications: authentication, digital signatures, multi-party computation, ...

Digital signature/cryptographic proof:

Connection to cryptography

> "Making sure certain things computationally hard, while keeping others easy."

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Many other applications: authentication, digital signatures, multi-party computation, ...

Digital signature/cryptographic proof:

- **Easy** for signer (on a laptop) to create signature

Connection to cryptography

> "Making sure certain things computationally hard, while keeping others easy."

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Many other applications: authentication, digital signatures, multi-party computation, ...

Digital signature/cryptographic proof:

- **Easy** for signer (on a laptop) to create signature
- **Hard** for even supercomputer to forge a signature

Connection to cryptography

> "Making sure certain things computationally hard, while keeping others easy."

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Many other applications: authentication, digital signatures, multi-party computation, ...

Digital signature/cryptographic proof:

- **Easy** for signer (on a laptop) to create signature
- **Hard** for even supercomputer to forge a signature
- **Easy** for recipient (on a laptop) to verify signature

Connection to cryptography

> "Making sure certain things computationally hard, while keeping others easy."

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Many other applications: authentication, digital signatures, multi-party computation, ...

Digital signature/cryptographic proof:

- **Easy** for signer (on a laptop) to create signature
- **Hard** for even supercomputer to forge a signature
- **Easy** for recipient (on a laptop) to verify signature

Connection to cryptography

"Making sure certain things computationally hard, while keeping others easy."

Encryption: should be hard in practice for eavesdropper to discover the secret message; easy for intended recipient.

Many other applications: authentication, digital signatures, multi-party computation, ...

Digital signature/cryptographic proof:

- **Easy** for signer (on a laptop) to create signature
- **Hard** for even supercomputer to forge a signature
- **Easy** for recipient (on a laptop) to verify signature

Our goal: a **"cryptographic proof of quantumness"**

Near-term verifiable quantum advantage

Trivial solution: Shor's algorithm

Near-term verifiable quantum advantage

Trivial solution: Shor's algorithm... but we want to do near-term!

Near-term verifiable quantum advantage

Trivial solution: Shor's algorithm... but we want to do near-term!

NISQ: Noisy Intermediate-Scale Quantum devices

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

Generically: seems hard.

The point of random circuits is that they don't have structure!

IQP

Example: sampling "IQP" circuits (products of Pauli X's)

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

Example: sampling "IQP" circuits (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 2009] Claim: Can hide a secret $\vec{s}$ in H, such that:

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

IQP

Example: sampling "IQP" circuits (products of Pauli X's)

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

[Shepherd, Bremner 2009] Claim: Can hide a secret $\vec{s}$ in H, such that:

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

For proof, collect many (unique) samples, and statistically establish that $p_{\vec{x}.\vec{s}} > 75\%$

Example: sampling "IQP" circuits (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 2009] Claim: Can hide a secret $\vec{s}$ in H, such that:

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

For proof, collect many (unique) samples, and statistically establish that $p_{\vec{x}.\vec{s}} > 75\%$

- **Easy** for quantum device to pass: yes

IQP

Example: sampling "IQP" circuits (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 2009] Claim: Can hide a secret $\vec{s}$ in H, such that:

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

For proof, collect many (unique) samples, and statistically establish that $p_{\vec{x}.\vec{s}} > 75\%$

- **Easy** for quantum device to pass: yes
- **Easy** for classical computer to verify: yes

IQP

Example: sampling "IQP" circuits (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 2009] Claim: Can hide a secret $\vec{s}$ in H, such that:

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

For proof, collect many (unique) samples, and statistically establish that $p_{\vec{x}.\vec{s}} > 75\%$

- **Easy** for quantum device to pass: yes
- **Easy** for classical computer to verify: yes
- **Hard** for classical computer to cheat: hopefully?

IQP

Example: sampling "IQP" circuits (products of Pauli X's)

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

[Shepherd, Bremner 2009] Claim: Can hide a secret $\vec{s}$ in H, such that:

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

For proof, collect many (unique) samples, and statistically establish that $p_{\vec{x}.\vec{s}} > 75\%$

- **Easy** for quantum device to pass: yes
- **Easy** for classical computer to verify: yes
- **Hard** for classical computer to cheat: hopefully?
  - Is it possible to simulate this class of circuits?

Example: sampling "IQP" circuits (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 2009] Claim: Can hide a secret $\vec{s}$ in H, such that:

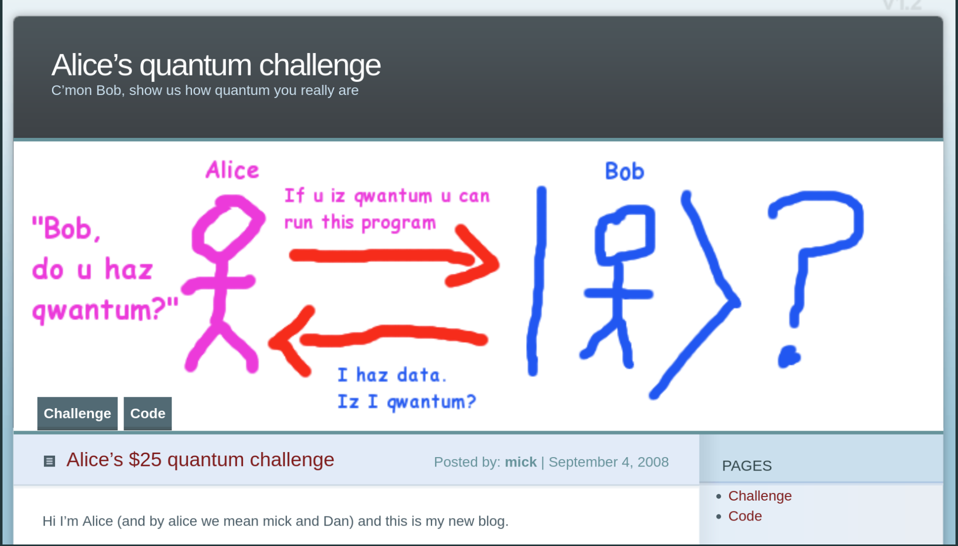
Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

For proof, collect many (unique) samples, and statistically establish that $p_{\vec{x}.\vec{s}} > 75\%$

- **Easy** for quantum device to pass: yes
- **Easy** for classical computer to verify: yes
- **Hard** for classical computer to cheat: hopefully?
  - Is it possible to simulate this class of circuits?
  - Is there some way to pass the test *without* simulating the circuit?

The $25 challenge



IQP: is it possible to simulate classically?

Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy

By Michael J. Bremner[1,*], Richard Jozsa[2] and Dan J. Shepherd[3]

[1] *Institut für Theoretische Physik, Leibniz Universität Hannover, Appelstrasse 2, Hannover 30167, Germany*
[2] *DAMTP, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, UK*
[3] *CESG, Hubble Road, Cheltenham GL51 0EX, UK*

PRL **117,** 080501 (2016) PHYSICAL REVIEW LETTERS week ending 19 AUGUST 2016

Average-Case Complexity Versus Approximate Simulation of Commuting Quantum Computations

Michael J. Bremner,[1,*] Ashley Montanaro,[2] and Dan J. Shepherd[3]

[1] *Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia*
[2] *School of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom*
[3] *CESG, Hubble Road, Cheltenham GL51 0EX, United Kingdom*

(Received 8 May 2015; revised manuscript received 9 June 2016; published 18 August 2016)

IQP: is it possible to simulate classically?

Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy

By Michael J. Bremner[1,*], Richard Jozsa[2] and Dan J. Shepherd[3]

[1] *Institut für Theoretische Physik, Leibniz Universität Hannover, Appelstrasse 2, Hannover 30167, Germany*
[2] *DAMTP, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, UK*
[3] *CESG, Hubble Road, Cheltenham GL51 0EX, UK*

PRL **117,** 080501 (2016) PHYSICAL REVIEW LETTERS week ending 19 AUGUST 2016

Average-Case Complexity Versus Approximate Simulation of Commuting Quantum Computations

Michael J. Bremner,[1,*] Ashley Montanaro,[2] and Dan J. Shepherd[3]

[1] *Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia*
[2] *School of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom*
[3] *CESG, Hubble Road, Cheltenham GL51 0EX, United Kingdom*

(Received 8 May 2015; revised manuscript received 9 June 2016; published 18 August 2016)

… and in practice, it seems to be infeasible for > 50 qubits…

IQP: is it possible to pass without simulating the circuit?

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

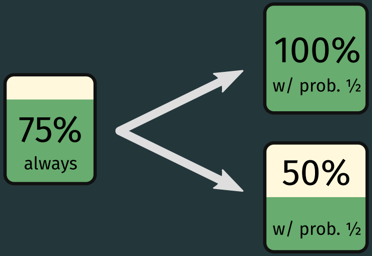
Quantum: $\sim$85% **Classical:** $\leq$75%

IQP: is it possible to pass without simulating the circuit?

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

Key: for a given H (and thus $\vec{s}$) one can classically generate sets of correlated samples.

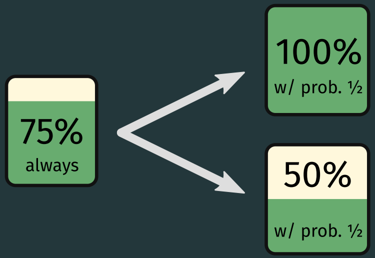


IQP: is it possible to pass without simulating the circuit?

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\lesssim$75%

Key: for a given H (and thus $\vec{s}$) one can classically generate sets of correlated samples.



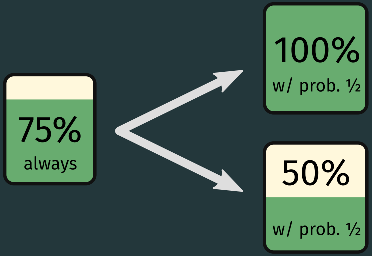
Q: why doesn't this immediately break the protocol?

IQP: is it possible to pass without simulating the circuit?

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

Key: for a given H (and thus $\vec{s}$) one can classically generate sets of correlated samples.



Q: why doesn't this immediately break the protocol?

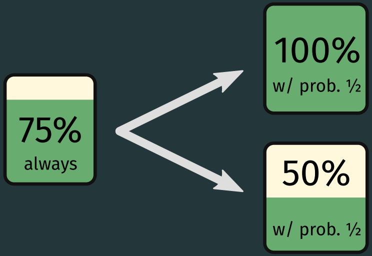
But...
In 100% case, get a system of equations for **s**!

IQP: is it possible to pass without simulating the circuit?

Fraction of measurement results with $\vec{x} \cdot \vec{s} = 0$:

Quantum: $\sim$85% **Classical:** $\leq$75%

Key: for a given H (and thus $\vec{s}$) one can classically generate sets of correlated samples.



Q: why doesn't this immediately break the protocol?

But...
In 100% case, get a system of equations for **s**!

With knowledge of $\vec{s}$, trivial to classically pass test.

Breaking the IQP protocol

Trying it against their verification code...

```
$ ./IQPwn challenge.dat
```

Trying it against their verification code...

```
$ ./IQPwn challenge.dat
Loading X-program at 'challenge.dat'...
Extracting secret key...
Generating samples...
Samples written to file 'response.dat'
$
```

Breaking the IQP protocol

Trying it against their verification code...

```
$ ./IQPwn challenge.dat
Loading X-program at 'challenge.dat'...
Extracting secret key...
Generating samples...
Samples written to file 'response.dat'
$ ./verify response.dat
```

Breaking the IQP protocol

Trying it against their verification code...

```
$ ./IQPwn challenge.dat
Loading X-program at 'challenge.dat'...
Extracting secret key...
Generating samples...
Samples written to file 'response.dat'
$ ./verify response.dat
Congratulations; you have what appears to be a
working quantum computer!
Dataset accepted as proof!
$ 
```

Near-term verifiable quantum advantage

NISQ: Noisy Intermediate-Scale Quantum devices

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

Making number theoretic problems less costly

Fully solving a problem like factoring is “overkill”

Making number theoretic problems less costly

Fully solving a problem like factoring is “overkill”

Can we demonstrate quantum *capability* without needing to solve such a hard problem?

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.

How can they use a red ball and green ball to convince you that they see color?

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.

How can they use a red ball and green ball to convince you that they see color?

without ever telling you the colors?

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.
How can they use a red ball and green ball to convince you that they see color?
without ever telling you the colors?

1. You show them one ball, then hide it behind your back

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.
How can they use a red ball and green ball to convince you that they see color?
without ever telling you the colors?

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

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.
How can they use a red ball and green ball to convince you that they see color?
without ever telling you the colors?

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

Impostor has 50% chance of passing—iterate for exponential certainty.

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.
How can they use a red ball and green ball to convince you that they see color?
without ever telling you the colors?

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

Impostor has 50% chance of passing—iterate for exponential certainty.

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

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.
How can they use a red ball and green ball to convince you that they see color?
without ever telling you the colors?

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

You (color blind) $\Leftrightarrow$ Classical verifier
Seeing color $\Leftrightarrow$ Quantum capability

Zero-knowledge proofs: differentiating colors

You are red/green colorblind, your friend is not.
How can they use a red ball and green ball to convince you that they see color?
without ever telling you the colors?

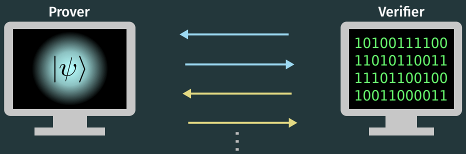
This constitutes a **zero-knowledge interactive proof**.

You (color blind) $\Leftrightarrow$ Classical verifier
Seeing color $\Leftrightarrow$ Quantum capability

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

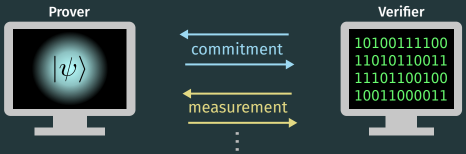
Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier



Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier

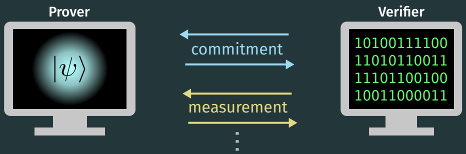


Round 1: Prover commits to holding a specific quantum state

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

By randomizing choice of basis and repeating interaction,
can ensure prover would respond correctly in *any* basis

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

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 $y = f(x_0) = f(x_1)$.

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

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 $y = f(x_0) = f(x_1)$.

Evaluate f on uniform superposition $\sum_x \|x\rangle \|f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick 2-to-1 function f
Measure 2^{nd} register as y	$\xrightarrow{\quad y \quad}$	Store y as commitment

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 $y = f(x_0) = f(x_1)$.

Evaluate f on uniform superposition $\sum_x \lvert x\rangle \lvert f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick 2-to-1 function f
Measure 2^{nd} 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

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

Function is actually 4-to-1 but collisions like $\{x, -x\}$ are trivial—set domain to integers in range $[0, N/2]$.

Trapdoor claw-free function example

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

Function is actually 4-to-1 but collisions like $\{x, -x\}$ are trivial—set domain to integers in range $[0, N/2]$.

Properties:

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

Trapdoor claw-free function example

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

Function is actually 4-to-1 but collisions like $\{x, -x\}$ are trivial—set domain to integers in range $[0, N/2]$.

Properties:

- **Claw-free:** Easy to compute p, q given a colliding pair—thus finding collisions is as hard as factoring. This is called a reduction
- **Trapdoor:** Function is easily inverted with knowledge of p, q

Trapdoor claw-free function example

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

Function is actually 4-to-1 but collisions like $\{x, -x\}$ are trivial—set domain to integers in range $[0, N/2]$.

Properties:

- **Claw-free:** Easy to compute p, q given a colliding pair—thus finding collisions is as hard as factoring. This is called a reduction
- **Trapdoor:** Function is easily inverted with knowledge of p, q

Example: $4^2 \equiv 11^2 \equiv 16 \pmod{35}$; and $11 - 4 = 7$

Brakerski, Christiano, Mahadev, Vazirani, Vidick ’18

Evaluate f on uniform superposition: $\sum_x |x\rangle |f(x)\rangle$ $\xleftarrow{\quad f \quad}$ Pick trapdoor claw-free function f

Measure 2nd register as y $\xrightarrow{\quad y \quad}$ Compute x_0, x_1 from y using trapdoor

arXiv:1804.00640

Brakerski, Christiano, Mahadev, Vazirani, Vidick '18

Evaluate f on uniform superposition: $\sum_x \|x\rangle \|f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick trapdoor claw-free function f
Measure 2$^{\text{nd}}$ register as y	$\xrightarrow{\quad y \quad}$	Compute x_0, x_1 from y using trapdoor
	$\xleftarrow{\text{basis}}$	Pick Z or X basis
Measure qubits of $\|x_0\rangle + \|x_1\rangle$ in given basis		
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

arXiv:1804.00640

Brakerski, Christiano, Mahadev, Vazirani, Vidick ’18

Evaluate f on uniform superposition: $\sum_x \|x\rangle \|f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick trapdoor claw-free function f
Measure 2nd register as y	$\xrightarrow{\quad y \quad}$	Compute x_0, x_1 from y using trapdoor
	$\xleftarrow{\text{basis}}$	Pick Z or X basis
Measure qubits of $\|x_0\rangle + \|x_1\rangle$ in given basis		
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Z basis: get x_0 or x_1

arXiv:1804.00640

Brakerski, Christiano, Mahadev, Vazirani, Vidick '18

Evaluate f on uniform superposition: $\sum_x |x\rangle\,|f(x)\rangle$ $\xleftarrow{\quad f \quad}$ Pick trapdoor claw-free function f

Measure 2nd register as y $\xrightarrow{\quad y \quad}$ Compute x_0, x_1 from y using trapdoor

$\xleftarrow{\quad \text{basis} \quad}$ Pick Z or X basis

Measure qubits of $|x_0\rangle + |x_1\rangle$ in given basis

$\xrightarrow{\quad \text{result} \quad}$ Validate result against x_0, x_1

Z basis: get x_0 or x_1

X basis: get some bitstring d, such that $d \cdot x_0 = d \cdot x_1$

arXiv:1804.00640

Brakerski, Christiano, Mahadev, Vazirani, Vidick '18

Evaluate f on uniform superposition: $\sum_x \|x\rangle \|f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick trapdoor claw-free function f
Measure 2nd register as y	$\xrightarrow{\quad y \quad}$	Compute x_0, x_1 from y using trapdoor
	$\xleftarrow{\text{basis}}$	Pick Z or X basis
Measure qubits of $\|x_0\rangle + \|x_1\rangle$ in given basis		
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Z basis: get x_0 or x_1

X basis: get some bitstring d, such that $d \cdot x_0 = d \cdot x_1$

Hardness of finding (x_0, x_1) does *not* imply hardness of measurement results!

arXiv:1804.00640

Brakerski, Christiano, Mahadev, Vazirani, Vidick ’18

Evaluate f on uniform superposition: $\sum_x |x\rangle\,|f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick trapdoor claw-free function f
Measure 2$^{\text{nd}}$ register as y	$\xrightarrow{\quad y \quad}$	Compute x_0, x_1 from y using trapdoor
	$\xleftarrow{\text{basis}}$	Pick Z or X basis
Measure qubits of $|x_0\rangle + |x_1\rangle$ in given basis		
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Hardness of finding (x_0, x_1) does *not* imply hardness of measurement results!

arXiv:1804.00640

Brakerski, Christiano, Mahadev, Vazirani, Vidick '18

		
Evaluate f on uniform superposition: $\sum_x \|x\rangle \|f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick trapdoor claw-free function f
Measure 2nd register as y	$\xrightarrow{\quad y \quad}$	Compute x_0, x_1 from y using trapdoor
	$\xleftarrow{\text{basis}}$	Pick Z or X basis
Measure qubits of $\|x_0\rangle + \|x_1\rangle$ in given basis		
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Hardness of finding (x_0, x_1) does *not* imply hardness of measurement results!

Protocol requires strong claw-free property:

For any x_0, hard to find even **a single bit** about x_1.

arXiv:1804.00640

Trapdoor claw-free functions

Function family	Trapdoor	Claw-free	Strong claw-free
Learning-with-Errors [1]	✓	✓	✓
Ring Learning-with-Errors [2]	✓	✓	✗
$x^2 \bmod N$ [3]	✓	✓	✗
Diffie-Hellman [3]	✓	✓	✗

[1] Brakerski, Christiano, Mahadev, Vidick, Vazirani '18 (arXiv:1804.00640)

[2] Brakerski, Koppula, Vazirani, Vidick '20 (arXiv:2005.04826)

[3] GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Trapdoor claw-free functions

Function family	Trapdoor	Claw-free	Strong claw-free
Learning-with-Errors [1]	✓	✓	✓
Ring Learning-with-Errors [2]	✓	✓	✗
$x^2 \bmod N$ [3]	✓	✓	✗
Diffie-Hellman [3]	✓	✓	✗

BKVV '20 removes need for strong claw-free property in the **random oracle model**. [2]

[1] Brakerski, Christiano, Mahadev, Vidick, Vazirani '18 (arXiv:1804.00640)

[2] Brakerski, Koppula, Vazirani, Vidick '20 (arXiv:2005.04826)

[3] GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Trapdoor claw-free functions

Function family	Trapdoor	Claw-free	Strong claw-free
Learning-with-Errors [1]	✓	✓	✓
Ring Learning-with-Errors [2]	✓	✓	✗
$x^2 \bmod N$ [3]	✓	✓	✗
Diffie-Hellman [3]	✓	✓	✗

BKVV '20 removes need for strong claw-free property in the **random oracle model**. [2]

Can we do the same in the **standard model**?

[1] Brakerski, Christiano, Mahadev, Vidick, Vazirani '18 (arXiv:1804.00640)

[2] Brakerski, Koppula, Vazirani, Vidick '20 (arXiv:2005.04826)

[3] GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Trapdoor claw-free functions

Function family	Trapdoor	Claw-free	Strong claw-free
Learning-with-Errors [1]	✓	✓	✓
Ring Learning-with-Errors [2]	✓	✓	✗
$x^2 \bmod N$ [3]	✓	✓	✗
Diffie-Hellman [3]	✓	✓	✗

BKVV '20 removes need for strong claw-free property in the **random oracle model**. [2]

Can we do the same in the **standard model**? Yes! [3]

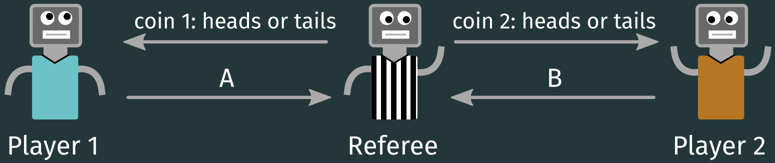
[1] Brakerski, Christiano, Mahadev, Vidick, Vazirani '18 (arXiv:1804.00640)

[2] Brakerski, Koppula, Vazirani, Vidick '20 (arXiv:2005.04826)

[3] GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Aside: the CHSH game (Bell test)

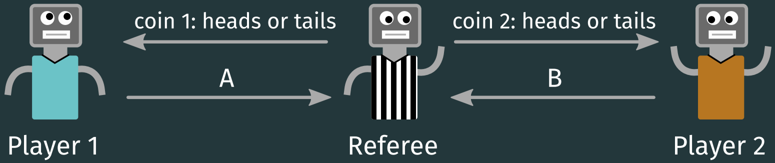
Cooperative two-player game; players can't communicate (non-local).



If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Aside: the CHSH game (Bell test)

Cooperative two-player game; players can't communicate (non-local).



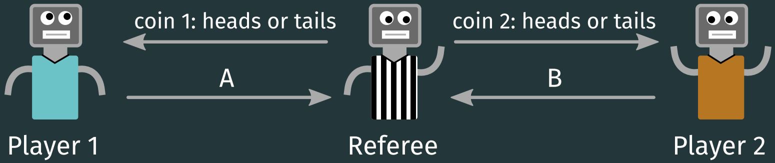
If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Classical optimal strategy: return equal values, hope you didn't both get heads. 75% success rate.

Can we do better with entanglement?

Aside: the CHSH game (Bell test)

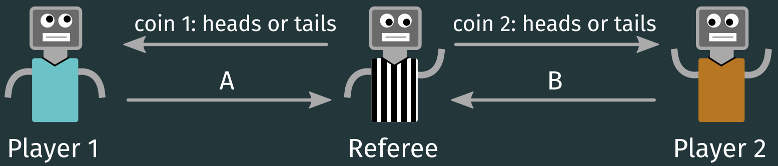
Cooperative two-player game; players can't communicate (non-local).



If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Consider the Bell pair: $|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle$

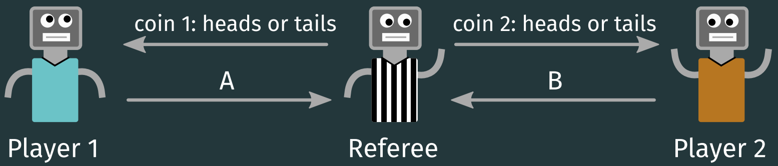
Aside: the CHSH game (Bell test)



If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Consider the Bell pair: $|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle = |\leftarrow\leftarrow\rangle + |\rightarrow\rightarrow\rangle = \cdots$

Aside: the CHSH game (Bell test)

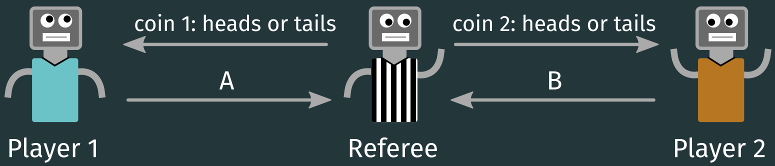


If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Consider the Bell pair: $|\psi\rangle = |{\uparrow\uparrow}\rangle + |{\downarrow\downarrow}\rangle = |{\leftarrow\leftarrow}\rangle + |{\rightarrow\rightarrow}\rangle = \cdots$

Aligned basis $\rightarrow$ same result; antialigned $\rightarrow$ opposite result!

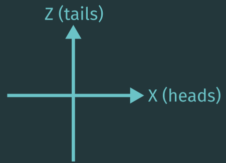
Aside: the CHSH game (Bell test)



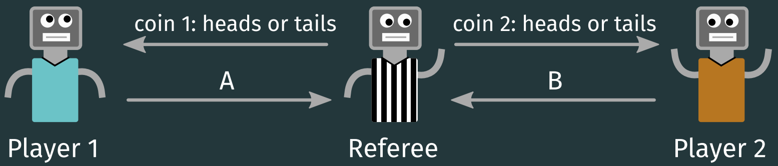
If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Consider the Bell pair: $|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle = |\leftarrow\leftarrow\rangle + |\rightarrow\rightarrow\rangle = \cdots$

Aligned basis $\rightarrow$ same result; antialigned $\rightarrow$ opposite result!



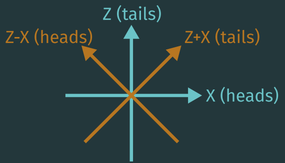
Aside: the CHSH game (Bell test)



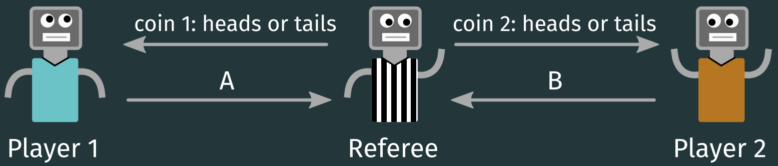
If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Consider the Bell pair: $|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle = |\leftarrow\leftarrow\rangle + |\rightarrow\rightarrow\rangle = \cdots$

Aligned basis $\rightarrow$ same result; antialigned $\rightarrow$ opposite result!



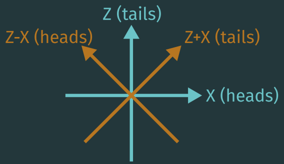
Aside: the CHSH game (Bell test)



If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Consider the Bell pair: $|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle = |\leftarrow\leftarrow\rangle + |\rightarrow\rightarrow\rangle = \cdots$

Aligned basis $\rightarrow$ same result; antialigned $\rightarrow$ opposite result!



Quantum: $\cos^2(\pi/8) \approx 85\%$

Classical: 75%

Brakerski, Christiano, Mahadev, Vazirani, Vidick ’18

Evaluate f on uniform superposition: $\sum_x \lvert x\rangle \lvert f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick trapdoor claw-free function f
Measure 2nd register as y	$\xrightarrow{\quad y \quad}$	Compute x_0, x_1 from y using trapdoor
Measure qubits of $\lvert x_0\rangle + \lvert x_1\rangle$ in given basis	$\xleftarrow{\text{basis}}$	Pick Z or X basis
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Brakerski, Christiano, Mahadev, Vazirani, Vidick '18

Evaluate f on uniform superposition: $\sum_x \|x\rangle \|f(x)\rangle$	$\xleftarrow{\quad f \quad}$	Pick trapdoor claw-free function f
Measure 2nd register as y	$\xrightarrow{\quad y \quad}$	Compute x_0, x_1 from y using trapdoor
Measure qubits of $\|x_0\rangle + \|x_1\rangle$ in given basis	$\xleftarrow{\text{basis}}$	Pick Z or X basis
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Replace X basis measurement with "single-qubit CHSH game"

Interactive measurement: computational Bell test

Two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

$\vdots$ $\qquad\qquad$ $\vdots$ $\qquad\qquad$ $\vdots$

$|x_0\rangle |x_0 \cdot r\rangle + |x_1\rangle |x_1 \cdot r\rangle$ $\xleftarrow{\quad r \quad}$ Pick random bitstring r

Measure all but ancilla in X basis $\xrightarrow{\quad d \quad}$

GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

$\vdots$ $\qquad\qquad$ $\vdots$ $\qquad\qquad$ $\vdots$

$|x_0\rangle |x_0 \cdot r\rangle + |x_1\rangle |x_1 \cdot r\rangle$ $\qquad \xleftarrow{\quad r \quad}$ $\qquad$ Pick random bitstring r

Measure all but ancilla in X basis $\qquad \xrightarrow{\quad d \quad}$

Now 1-qubit state: $|0\rangle$ or $|1\rangle$ if $x_0 \cdot r = x_1 \cdot r$, otherwise $|+\rangle$ or $|-\rangle$.

GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Interactive measurement: computational Bell test

Two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

$|x_0\rangle\,|x_0 \cdot r\rangle + |x_1\rangle\,|x_1 \cdot r\rangle$ $\xleftarrow{\quad r \quad}$ Pick random bitstring r

Measure all but ancilla in X basis $\xrightarrow{\quad d \quad}$

Now 1-qubit state: $|0\rangle$ or $|1\rangle$ if $x_0 \cdot r = x_1 \cdot r$, otherwise $|+\rangle$ or $|-\rangle$. Polarization hidden via:

Cryptographic secret (here) $\Leftrightarrow$ Non-communication (Bell test)

GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Interactive measurement: computational Bell test

Two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

$\vdots$	$\vdots$	$\vdots$
$\lvert x_0\rangle \lvert x_0 \cdot r\rangle + \lvert x_1\rangle \lvert x_1 \cdot r\rangle$	$\xleftarrow{\quad r \quad}$	Pick random bitstring r
Measure all but ancilla in X basis	$\xrightarrow{\quad d \quad}$	
Measure qubit in basis	$\xleftarrow{\text{basis}}$	Pick $(Z+X)$ or $(Z-X)$ basis
	$\xrightarrow{\text{result}}$	Validate against r, x_0, x_1, d

GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Interactive measurement: computational Bell test

Two-step process: “condense” x_0, x_1 into a single qubit, and then do a “Bell test.”

$\vdots$	$\vdots$	$\vdots$
$\|x_0\rangle \|x_0 \cdot r\rangle + \|x_1\rangle \|x_1 \cdot r\rangle$	$\xleftarrow{\quad r \quad}$	Pick random bitstring r
Measure all but ancilla in X basis	$\xrightarrow{\quad d \quad}$	
Measure qubit in basis	$\xleftarrow{\text{basis}}$	Pick $(Z+X)$ or $(Z-X)$ basis
	$\xrightarrow{\text{result}}$	Validate against r, x_0, x_1, d

This protocol can use any trapdoor claw-free function!

GDKM, Choi, Vazirani, Yao ’21 (arXiv:2104.00687)

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_Z: Success rate for Z basis measurement.

p_{Bell}: Success rate when performing Bell-type measurement.

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_Z: Success rate for Z basis measurement.

p_{Bell}: Success rate when performing Bell-type measurement.

Under assumption of claw-free function:

Classical bound: $p_Z + 4p_{\text{Bell}} \lesssim 4$

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_Z: Success rate for Z basis measurement.

p_{Bell}: Success rate when performing Bell-type measurement.

Under assumption of claw-free function:

Classical bound: $p_Z + 4p_{\text{Bell}} \lesssim 4$

Ideal quantum: $p_Z = 1, p_{\text{Bell}} = \cos^2(\pi/8)$

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_Z: Success rate for Z basis measurement.

p_{Bell}: Success rate when performing Bell-type measurement.

Under assumption of claw-free function:

Classical bound: $p_Z + 4p_{\text{Bell}} \lesssim 4$

Ideal quantum: $p_Z = 1, p_{\text{Bell}} = \cos^2(\pi/8)$

$p_Z + 4p_{\text{Bell}} = 3 + \sqrt{2} \approx \mathbf{4.414}$

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_Z: Success rate for Z basis measurement.

p_{Bell}: Success rate when performing Bell-type measurement.

Under assumption of claw-free function:

Classical bound: $p_Z + 4p_{\text{Bell}} \lesssim 4$

Ideal quantum: $p_Z = 1, p_{\text{Bell}} = \cos^2(\pi/8)$

$p_Z + 4p_{\text{Bell}} = 3 + \sqrt{2} \approx \mathbf{4.414}$

Note: Let $p_Z = 1$. Then for p_{Bell}:

Classical bound 75%, ideal quantum $\sim 85\%$. Same as regular Bell test!

GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Overview: efficiently verifiable quantum advantage protocol

- Existing experiments (e.g. random circuits) not verifiable at scale; classical hardness less well established

Overview: efficiently verifiable quantum advantage protocol

- Existing experiments (e.g. random circuits) not verifiable at scale; classical hardness less well established
- Shor's alg. (and others) verifiable, but not feasible on near-term devices

Overview: efficiently verifiable quantum advantage protocol

- Existing experiments (e.g. random circuits) not verifiable at scale; classical hardness less well established
- Shor's alg. (and others) verifiable, but not feasible on near-term devices
- **Idea:** use zero-knowledge interactive proof to achieve hardness and verifiability of factoring, without full machinery of Shor

Overview: efficiently verifiable quantum advantage protocol

- Existing experiments (e.g. random circuits) not verifiable at scale; classical hardness less well established
- Shor's alg. (and others) verifiable, but not feasible on near-term devices
- **Idea:** use zero-knowledge interactive proof to achieve hardness and verifiability of factoring, without full machinery of Shor
- **Result:** new protocol that allows proof of quantumness using any trapdoor claw-free function, including $x^2 \bmod N$

Overview: efficiently verifiable quantum advantage protocol

- Existing experiments (e.g. random circuits) not verifiable at scale; classical hardness less well established
- Shor's alg. (and others) verifiable, but not feasible on near-term devices
- **Idea:** use zero-knowledge interactive proof to achieve hardness and verifiability of factoring, without full machinery of Shor
- **Result:** new protocol that allows proof of quantumness using any trapdoor claw-free function, including $x^2 \bmod N$

Asymptotically: evaluating $x^2 \bmod N$ requires $\mathcal{O}(n \log n)$ gates; $a^x \bmod N$ in Shor requires $\mathcal{O}(n^2 \log n)$

(can also use other TCFs, and other optimizations...)

Moving towards efficiently-verifiable quantum advantage in the near term

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits
- Recent first implementations by experiments! [1]

[1] GDKM, D. Zhu, et al. '21 (arXiv:2112.05156)

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits
- Recent first implementations by experiments! [1]

Fidelity (without error correction)

[1] GDKM, D. Zhu, et al. '21 (arXiv:2112.05156)

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits
- Recent first implementations by experiments! [1]

Fidelity (without error correction)

- Need to pass classical threshold

[1] GDKM, D. Zhu, et al. '21 (arXiv:2112.05156)

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits
- Recent first implementations by experiments! [1]

Fidelity (without error correction)

- Need to pass classical threshold
- Postselection scheme enables passing with ϵ circuit fidelity [2]

[1] GDKM, D. Zhu, et al. '21 (arXiv:2112.05156)
[2] GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits
- Recent first implementations by experiments! [1]

Fidelity (without error correction)

- Need to pass classical threshold
- Postselection scheme enables passing with ϵ circuit fidelity [2]

Circuit sizes

[1] GDKM, D. Zhu, et al. '21 (arXiv:2112.05156)

[2] GDKM, Choi, Vazirani, Yao '21 (arXiv:2104.00687)

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits
- Recent first implementations by experiments! [1]

Fidelity (without error correction)

- Need to pass classical threshold
- Postselection scheme enables passing with ϵ circuit fidelity [2]

Circuit sizes

- Removing need for strong claw-free property allows use of “easier” functions

[1] GDKM, D. Zhu, et al. ’21 (arXiv:2112.05156)
[2] GDKM, Choi, Vazirani, Yao ’21 (arXiv:2104.00687)

Moving towards efficiently-verifiable quantum advantage in the near term

Interaction

- Mid-circuit measurement: need to measure subsystem while maintaining coherence on other qubits
- Recent first implementations by experiments! [1]

Fidelity (without error correction)

- Need to pass classical threshold
- Postselection scheme enables passing with ϵ circuit fidelity [2]

Circuit sizes

- Removing need for strong claw-free property allows use of “easier” functions
- Measurement-based uncomputation scheme [2]

[1] GDKM, D. Zhu, et al. ’21 (arXiv:2112.05156)

[2] GDKM, Choi, Vazirani, Yao ’21 (arXiv:2104.00687)

Intermediate (mid-circuit) measurements

Principle of delayed measurement: delaying all measurements to the end of a circuit doesn't affect the measurement statistics.

Intermediate (mid-circuit) measurements

Principle of delayed measurement: delaying all measurements to the end of a circuit doesn't affect the measurement statistics.

Q: Why is mid-circuit measurement necessary for these protocols?

Intermediate (mid-circuit) measurements

Principle of delayed measurement: delaying all measurements to the end of a circuit doesn't affect the measurement statistics.

Q: Why is mid-circuit measurement necessary for these protocols?

Other applications of mid-circuit measurement:

- Quantum error correction
- Quantum machine learning (QCNN)
- ...

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

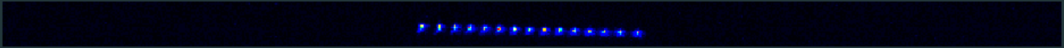
First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\to$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

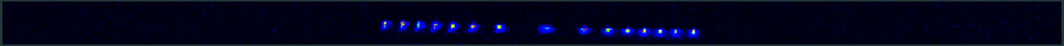


Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:



Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

Trapped Ion Quantum Information lab at U. Maryland ($\to$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

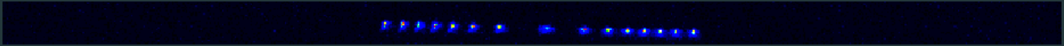
Partial measurement:

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\rightarrow$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

Partial measurement:

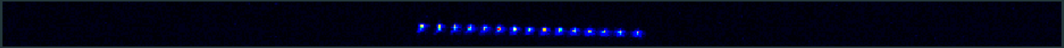


Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland ($\to$ Duke)

First demonstration of these protocols, in trapped ions! (arXiv:2112.05156)

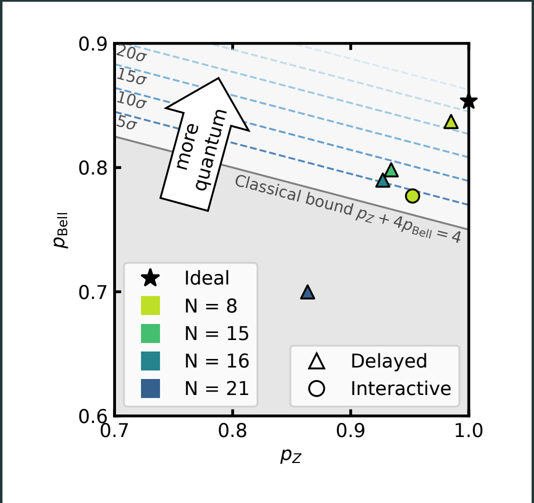
Partial measurement:



Experimental results for $f(x) = x^2 \bmod N$

Up and **right** is stronger evidence of quantumness

GDKM, D. Zhu, et al. (arXiv:2112.05156)



Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

- Preliminary implementation of $x^2 \mod N$ at scale has depth 10^5—optimize it!

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

- Preliminary implementation of $x^2 \bmod N$ at scale has depth 10^5—optimize it!
- Co-design circuits for specific hardware (Rydberg implementation in paper)

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

- Preliminary implementation of $x^2 \bmod N$ at scale has depth 10^5—optimize it!
- Co-design circuits for specific hardware (Rydberg implementation in paper)
- $x^2 \bmod N$ requires at minimum 500-1000 qubits for classical hardness—search for new claw-free functions?

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

- Preliminary implementation of $x^2 \bmod N$ at scale has depth 10^5—optimize it!
- Co-design circuits for specific hardware (Rydberg implementation in paper)
- $x^2 \bmod N$ requires at minimum 500-1000 qubits for classical hardness—search for new claw-free functions?

Improving the protocol itself:

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

- Preliminary implementation of $x^2 \bmod N$ at scale has depth 10^5—optimize it!
- Co-design circuits for specific hardware (Rydberg implementation in paper)
- $x^2 \bmod N$ requires at minimum 500-1000 qubits for classical hardness—search for new claw-free functions?

Improving the protocol itself:

- Remove trapdoor—symmetric key/hash-based cryptography [arXiv:2204.02063]

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

- Preliminary implementation of $x^2 \bmod N$ at scale has depth 10^5—optimize it!
- Co-design circuits for specific hardware (Rydberg implementation in paper)
- $x^2 \bmod N$ requires at minimum 500-1000 qubits for classical hardness—search for new claw-free functions?

Improving the protocol itself:

- Remove trapdoor—symmetric key/hash-based cryptography [arXiv:2204.02063]
- Explore other protocols (verifiable sampling with good security?)

References + further reading

Numbers below are arXiv IDs; go to arxiv.org/abs/xxxx.xxxxx

Proofs of quantumness

- IQP sampling protocol [0809.0847]
- Breaking IQP protocol [1912.05547]
- First interactive proof based on trapdoor claw-free functions [1804.00640]
- Removing assumptions via random oracles [2005.04826]
- Removing assumptions via computational Bell test [2104.00687]
- Single-prover proofs from any multi-prover quantum game [2203.15877]
- Proofs using only random oracles [2204.02063]

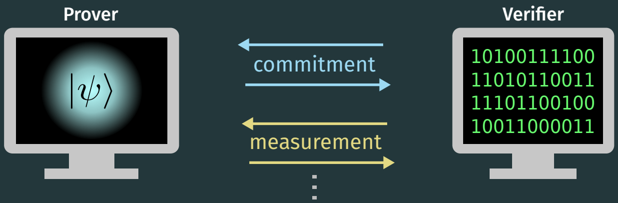
Other applications of quantum interactive proofs

- Certifiable quantum randomness [1804.00640]
- Remote state preparation [1904.06320]
- Verification of arbitrary quantum computations (!) [1804.01082]

Feel free to email me! Greg Kahanamoku-Meyer; gkm@berkeley.edu

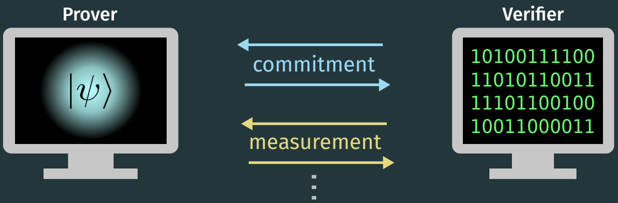
Backup!

Hardness proof: rewinding



From a "proof of hardness" perspective:

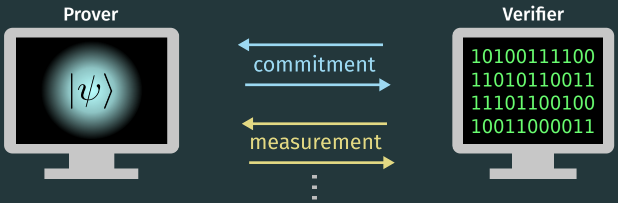
Hardness proof: rewinding



From a "proof of hardness" perspective:

- **Classical** cheater can be "rewound"
  - Save state of prover after first round of interaction
  - Extract measurement results in all choices of basis

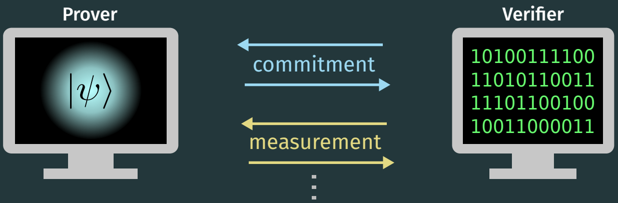
Hardness proof: rewinding



From a “proof of hardness” perspective:

- Classical cheater can be “rewound”
  - Save state of prover after first round of interaction
  - Extract measurement results in all choices of basis
- Quantum prover’s measurements are irreversible

Hardness proof: rewinding



From a "proof of hardness" perspective:

- **Classical** cheater can be "rewound"
  - Save state of prover after first round of interaction
  - Extract measurement results in all choices of basis
- **Quantum** prover's measurements are irreversible

"Rewinding" proof of hardness doesn't go through for quantum prover—can even use functions that are quantum claw-free!

Technique: postselection

How to deal with high fidelity requirement? Naively need $\sim 83\%$ overall circuit fidelity to pass.

Technique: postselection

How to deal with high fidelity requirement? Naively need $\sim 83\%$ overall circuit fidelity to pass.

A prover holding $(|x_0\rangle + |x_1\rangle)\,|y\rangle$ with ϵ phase coherence passes!

Technique: postselection

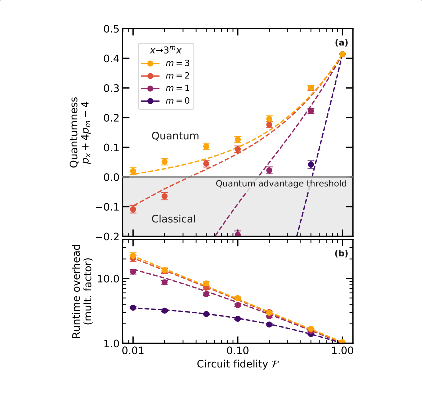
How to deal with high fidelity requirement? Naively need $\sim 83\%$ overall circuit fidelity to pass.

A prover holding $(|x_0\rangle + |x_1\rangle)|y\rangle$ with ϵ phase coherence passes!

When we generate $\sum_x |x\rangle |f(x)\rangle$, **add redundancy to $f(x)$, for bit flip error detection!**

Technique: postselection

How to deal with high fidelity requirement? Naively need $\sim 83\%$ overall circuit fidelity to pass.



Numerical results for $x^2 \bmod N$ with $\log N = 512$ bits.

Here: make transformation $x^2 \bmod N \Rightarrow (kx)^2 \bmod k^2 N$

Improving circuit sizes

Most demanding step in all these protocols: evaluating TCF

$$\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$$

Improving circuit sizes

Most demanding step in all these protocols: evaluating TCF

$$\mathcal{U}_f |x\rangle |0^{\otimes n}\rangle = |x\rangle |f(x)\rangle$$

Getting rid of strong claw-free property helps!

$x^2 \mod N$ and Ring-LWE have classical circuits as fast as $\mathcal{O}(n \log n)$...

Improving circuit sizes

Most demanding step in all these protocols: evaluating TCF

$$\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$$

Getting rid of strong claw-free property helps!

$x^2 \bmod N$ and Ring-LWE have classical circuits as fast as $\mathcal{O}(n \log n)$...

but they are recursive and hard to make reversible.

Improving circuit sizes

Most demanding step in all these protocols: evaluating TCF

$$\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$$

Getting rid of strong claw-free property helps!

$x^2 \mod N$ and Ring-LWE have classical circuits as fast as $\mathcal{O}(n \log n)$...

but they are recursive and hard to make reversible.

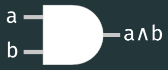
Protocol allows us to make circuits irreversible!

Technique: taking out the garbage

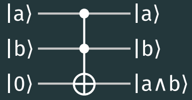
Goal: $\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity



Classical AND



Quantum AND (Toffoli)

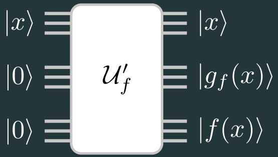
Technique: taking out the garbage

Goal: $\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity

Let $\mathcal{U}'_f$ be a unitary generating garbage bits $g_f(x)$:



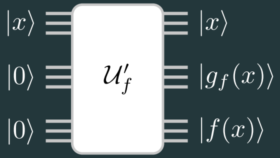
Technique: taking out the garbage

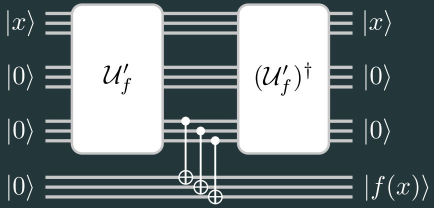
Goal: $\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity

Let $\mathcal{U}'_f$ be a unitary generating garbage bits $g_f(x)$:





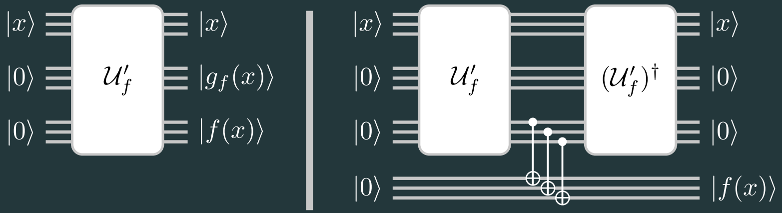
Technique: taking out the garbage

Goal: $\mathcal{U}_f \, |x\rangle \, |0^{\otimes n}\rangle = |x\rangle \, |f(x)\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity

Let $\mathcal{U}'_f$ be a unitary generating garbage bits $g_f(x)$:



Lots of time and space overhead!

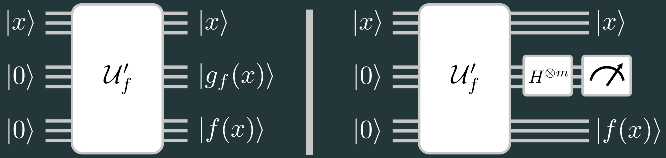
Technique: taking out the garbage

Goal: $\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity

Let $\mathcal{U}_f'$ be a unitary generating garbage bits $g_f(x)$:



Can we "measure them away" instead?

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in X basis, get some string h. End up with state:

$$\sum_x (-1)^{h \cdot g_f(x)} |x\rangle |f(x)\rangle$$

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in X basis, get some string h. End up with state:

$$\sum_x (-1)^{h \cdot g_f(x)} |x\rangle |f(x)\rangle$$

In general useless: unique phase $(-1)^{h \cdot g_f(x)}$ on every term.

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in X basis, get some string h. End up with state:

$$\sum_x (-1)^{h \cdot g_f(x)} |x\rangle |f(x)\rangle$$

In general useless: unique phase $(-1)^{h \cdot g_f(x)}$ on every term.

But after collapsing onto a single output:

$$\left[(-1)^{h \cdot g_f(x_0)} |x_0\rangle + (-1)^{h \cdot g_f(x_1)} |x_1\rangle\right] |y\rangle$$

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in X basis, get some string h. End up with state:

$$\sum_x (-1)^{h \cdot g_f(x)} |x\rangle |f(x)\rangle$$

In general useless: unique phase $(-1)^{h \cdot g_f(x)}$ on every term.

But after collapsing onto a single output:

$$[(-1)^{h \cdot g_f(x_0)} |x_0\rangle + (-1)^{h \cdot g_f(x_1)} |x_1\rangle] |y\rangle$$

Verifier can efficiently compute $g_f(\cdot)$ for these two terms!

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in X basis, get some string h. End up with state:

$$\sum_x (-1)^{h \cdot g_f(x)} |x\rangle |f(x)\rangle$$

In general useless: unique phase $(-1)^{h \cdot g_f(x)}$ on every term.

But after collapsing onto a single output:

$$[(-1)^{h \cdot g_f(x_0)} |x_0\rangle + (-1)^{h \cdot g_f(x_1)} |x_1\rangle] |y\rangle$$

Verifier can efficiently compute $g_f(\cdot)$ for these two terms!

Can directly convert classical circuits to quantum!

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in X basis, get some string h. End up with state:

$$\sum_x (-1)^{h \cdot g_f(x)} |x\rangle |f(x)\rangle$$

In general useless: unique phase $(-1)^{h \cdot g_f(x)}$ on every term.

But after collapsing onto a single output:

$$\left[(-1)^{h \cdot g_f(x_0)} |x_0\rangle + (-1)^{h \cdot g_f(x_1)} |x_1\rangle\right] |y\rangle$$

Verifier can efficiently compute $g_f(\cdot)$ for these two terms!

Can directly convert classical circuits to quantum!
1024-bit $x^2 \bmod N$ in depth 10^5 (and can be improved?)

IQP circuits [Shepherd and Bremner, '08]

Consider a matrix $P \in \{0,1\}^{k \times n}$ and "action" θ.

IQP circuits [Shepherd and Bremner, '08]

Consider a matrix $P \in \{0,1\}^{k \times n}$ and "action" θ.

Let $H = \sum_i \prod_j X_j^{P_{ij}}$.

Example:

$$H = X_0 X_1 X_3 + X_1 X_2 X_4 X_5 + \cdots \tag{2}$$

IQP circuits [Shepherd and Bremner, '08]

Consider a matrix $P \in \{0,1\}^{k \times n}$ and "action" θ.

Let $H = \sum_i \prod_j X_j^{P_{ij}}$.

Example:

$$H = X_0 X_1 X_3 + X_1 X_2 X_4 X_5 + \cdots \tag{2}$$

Distribution of sampling result X:

$$\Pr[X = x] = \left| \left\langle x \,\middle|\, e^{-iH\theta} \,\middle|\, 0 \right\rangle \right|^2 \tag{3}$$

IQP circuits [Shepherd and Bremner, '08]

Consider a matrix $P \in \{0,1\}^{k \times n}$ and "action" θ.

Let $H = \sum_i \prod_j X_j^{P_{ij}}$.

Example:

$$H = X_0X_1X_3 + X_1X_2X_4X_5 + \cdots \tag{2}$$

Distribution of sampling result X:

$$\Pr[X = x] = \left|\left\langle x \,\middle|\, e^{-iH\theta} \,\middle|\, 0 \right\rangle\right|^2 \tag{3}$$

Bremner, Jozsa, Shepherd '11: classically sampling worst-case IQP circuits would collapse polynomial heirarchy

Bremner, Montanaro, Shepherd '16: average case is likely hard as well

IQP proof of quantumness [Shepherd and Bremner, '08]

Let $\theta = \pi/8$, and s (secret) and P have the form:

$$P = \begin{bmatrix} \mathrm{G} \\ \hline \mathrm{R} \end{bmatrix}$$

$G^\intercal$ is generator of Quadratic Residue code, R random.

IQP proof of quantumness [Shepherd and Bremner, '08]

Let $\theta = \pi/8$, and s (secret) and P have the form:

$$P = \left[\begin{array}{c} G \\ \hline R \end{array}\right] \qquad Ps = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$G^\intercal$ is generator of Quadratic Residue code, R random.

$$\Pr[X^\intercal \cdot s = 0] = \mathop{\mathbb{E}}_{x}\left[\cos^2\left(\frac{\pi}{8}(1 - 2\mathrm{wt}(Gx))\right)\right]$$

IQP proof of quantumness [Shepherd and Bremner, '08]

Let $\theta = \pi/8$, and s (secret) and P have the form:

$$P = \begin{bmatrix} G \\ \hline R \end{bmatrix} \qquad Ps = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$G^\intercal$ is generator of Quadratic Residue code, R random.

$$\Pr[X^\intercal \cdot s = 0] = \mathop{\mathbb{E}}_{x}\left[\cos^2\left(\frac{\pi}{8}(1 - 2\mathrm{wt}(Gx))\right)\right]$$

QR code: codewords have $\mathrm{wt}(c) \bmod 4 \in \{0, -1\}$

IQP proof of quantumness [Shepherd and Bremner, '08]

Let $\theta = \pi/8$, and s (secret) and P have the form:

$$P = \left[\begin{array}{c} \text{G} \\ \hline \text{R} \end{array}\right] \qquad \boldsymbol{Ps} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$G^\intercal$ is generator of Quadratic Residue code, R random.

$$\Pr[X^\intercal \cdot \boldsymbol{s} = 0] = \cos^2\left(\frac{\pi}{8}\right) \approx 0.85$$

QR code: codewords have $\mathrm{wt}(\boldsymbol{c}) \bmod 4 \in \{0, -1\}$

IQP: Hiding s

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$

Best classical: $\Pr[Y^\intercal \cdot s = 0] = ?$

$$P = \begin{bmatrix} G \\ \hline R \end{bmatrix} \qquad P\mathbf{s} = \begin{bmatrix} 1\\1\\1\\1\\1\\1\\0\\0\\0\\0\\0\\0 \end{bmatrix}$$

IQP: Hiding s

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^\intercal \cdot s = 0] = ?$

$$P = \begin{bmatrix} G \\ \hline R \end{bmatrix} \qquad Ps = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{permute rows, Gauss-Jordan columns}} P'\mathbf{s}' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Scrambling preserves quantum success rate.

IQP: Hiding s

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^\intercal \cdot s = 0] = ?$

$$P = \begin{bmatrix} G \\ \hline R \end{bmatrix} \qquad P\mathbf{s} = \begin{bmatrix} 1\\1\\1\\1\\1\\1\\0\\0\\0\\0\\0\\0 \end{bmatrix} \quad \xrightarrow{\text{permute rows, Gauss-Jordan columns}} \quad P'\mathbf{s'} = \begin{bmatrix} 1\\0\\0\\1\\0\\1\\0\\1\\0\\0\\1\\1 \end{bmatrix}$$

Scrambling preserves quantum success rate.

Conjecture [SB '08]: Scrambling P cryptographically hides G (and equivalently $\mathbf{s}$)

IQP: Classical strategy

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Assuming s hidden, can classical do better than 0.5? Try to take advantage properties of embedded code.

IQP: Classical strategy

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Assuming s hidden, can classical do better than 0.5? **Try to take advantage properties of embedded code.**

Consider choosing random $d \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = 1}} p$$

IQP: Classical strategy

Quantum: $\Pr[X^{\intercal} \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^{\intercal} \cdot s = 0] \stackrel{?}{=} 0.5$

Assuming s hidden, can classical do better than 0.5? **Try to take advantage properties of embedded code.**

Consider choosing random $d \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = 1}} p$$

Then:

$$y \cdot s = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = 1}} p \cdot s \pmod 2$$

IQP: Classical strategy

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Assuming s hidden, can classical do better than 0.5? **Try to take advantage properties of embedded code.**

Consider choosing random $d \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = 1}} p$$

Then:

$$y \cdot s = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot s = 1}} 1 \pmod 2$$

IQP: Classical strategy

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Assuming s hidden, can classical do better than 0.5? **Try to take advantage properties of embedded code.**

Consider choosing random $d \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = 1}} p$$

Then:

$$y \cdot s = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot s = 1}} p \cdot d \pmod 2$$

IQP: Classical strategy

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Best classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Assuming s hidden, can classical do better than 0.5? **Try to take advantage properties of embedded code.**

Consider choosing random $d \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = 1}} p$$

Then:

$$y \cdot s = \mathrm{wt}(Gd) \pmod 2$$

QR code codewords are 50% even parity, 50% odd parity.

IQP: Classical strategy [SB '08]

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Consider choosing random $d, e \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot e = 1}} p$$

IQP: Classical strategy [SB '08]

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Consider choosing random $d, e \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot e = 1}} p$$

Then:

IQP: Classical strategy [SB '08]

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Consider choosing random $d, e \stackrel{\$}{\leftarrow} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot e = 1}} p$$

Then:

$$y \cdot s = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot e = 1}} p \cdot s \pmod 2$$

IQP: Classical strategy [SB '08]

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Consider choosing random $d, e \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot e = 1}} p$$

Then:

$$y \cdot s = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot s = 1}} (p \cdot d)(p \cdot e) \pmod 2$$

IQP: Classical strategy [SB '08]

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Classical: $\Pr[Y^\intercal \cdot s = 0] \stackrel{?}{=} 0.5$

Consider choosing random $\boldsymbol{d}, \boldsymbol{e} \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot e = 1}} p$$

Then:

$$y \cdot s = (G\boldsymbol{d}) \cdot (G\boldsymbol{e}) \pmod 2$$

Fact: $(G\boldsymbol{d}) \cdot (G\boldsymbol{e}) = 1$ iff $G\boldsymbol{d}$, $G\boldsymbol{e}$ both have odd parity.

IQP: Classical strategy [SB '08]

Quantum: $\Pr[X^\intercal \cdot s = 0] \approx 0.85$
Classical: $\Pr[Y^\intercal \cdot s = 0] = 0.75$

Consider choosing random $d, e \xleftarrow{\$} \{0,1\}^n$, and letting

$$y = \sum_{\substack{p \in \mathrm{rows}(P) \\ p \cdot d = p \cdot e = 1}} p$$

Then:

$$y \cdot s = (Gd) \cdot (Ge) \pmod 2$$

Fact: $(Gd) \cdot (Ge) = 1$ iff Gd, Ge both have odd parity.

Thus $y \cdot s = 0$ with probability $3/4$!

IQP: Can we do better classically? [GDKM ’19 arXiv:1912.05547]

Key: Correlate samples to attack the key s

IQP: Can we do better classically? [GDKM '19 arXiv:1912.05547]

Key: Correlate samples to attack the key s

Consider choosing one random $\boldsymbol{d} \xleftarrow{\$} \{0,1\}^n$, held constant over many different $\boldsymbol{e}_i \xleftarrow{\$} \{0,1\}^n$

$$\boldsymbol{y}_i = \sum_{\substack{p \in \mathrm{rows}(P) \\ \boldsymbol{p} \cdot \boldsymbol{d} = \boldsymbol{p} \cdot \boldsymbol{e}_i = 1}} \boldsymbol{p}$$

$\boldsymbol{y}_i \cdot s = 1$ iff $G\boldsymbol{d}$, $G\boldsymbol{e}_i$ both have odd parity.

IQP: Can we do better classically? [GDKM '19 arXiv:1912.05547]

Key: Correlate samples to attack the key $\mathbf{s}$

Consider choosing one random $\mathbf{d} \xleftarrow{\$} \{0,1\}^n$, held constant over many different $\mathbf{e}_i \xleftarrow{\$} \{0,1\}^n$

$$\mathbf{y}_i = \sum_{\substack{p \in \mathrm{rows}(P) \\ \mathbf{p}\cdot\mathbf{d} = \mathbf{p}\cdot\mathbf{e}_i = 1}} \mathbf{p}$$

$\mathbf{y}_i \cdot s = 1$ iff $G\mathbf{d}$, $G\mathbf{e}_i$ both have odd parity.

$G\mathbf{d}$ has even parity $\Rightarrow$ *all* $\mathbf{y}_i \cdot s = 0$

IQP: Can we do better classically? [GDKM '19 arXiv:1912.05547]

Key: Correlate samples to attack the key $\boldsymbol{s}$

Consider choosing one random $\boldsymbol{d} \xleftarrow{\$} \{0,1\}^n$, held constant over many different $\boldsymbol{e}_i \xleftarrow{\$} \{0,1\}^n$

$$\boldsymbol{y}_i = \sum_{\substack{p \in \mathrm{rows}(P) \\ \boldsymbol{p}\cdot\boldsymbol{d}=\boldsymbol{p}\cdot\boldsymbol{e}_i=1}} \boldsymbol{p}$$

$\boldsymbol{y}_i \cdot s = 1$ iff $G\boldsymbol{d}$, $G\boldsymbol{e}_i$ both have odd parity.

$G\boldsymbol{d}$ has even parity $\Rightarrow$ *all* $\boldsymbol{y}_i \cdot s = 0$

Let y_i form rows of a matrix M, such that $M\boldsymbol{s} = \boldsymbol{0}$

IQP: Can we do better classically? [GDKM '19 arXiv:1912.05547]

Key: Correlate samples to attack the key $\mathbf{s}$

Consider choosing one random $\mathbf{d} \xleftarrow{\$} \{0,1\}^n$, held constant over many different $\mathbf{e}_i \xleftarrow{\$} \{0,1\}^n$

$$\mathbf{y}_i = \sum_{\substack{p \in \mathrm{rows}(P) \\ \mathbf{p}\cdot\mathbf{d}=\mathbf{p}\cdot\mathbf{e}_i=1}} \mathbf{p}$$

$\mathbf{y}_i \cdot s = 1$ iff $G\mathbf{d}$, $G\mathbf{e}_i$ both have odd parity.

$G\mathbf{d}$ has even parity $\Rightarrow$ *all* $\mathbf{y}_i \cdot s = 0$

Let y_i form rows of a matrix M, such that $M\mathbf{s} = \mathbf{0}$

Can solve for $\mathbf{s}$! ... If M has high rank.

IQP: Can we do better classically? [GDKM '19 arXiv:1912.05547]

Key: Correlate samples to attack the key $\mathbf{s}$

Consider choosing one random $\mathbf{d} \xleftarrow{\$} \{0,1\}^n$, held constant over many different $\mathbf{e}_i \xleftarrow{\$} \{0,1\}^n$

$$\mathbf{y}_i = \sum_{\substack{p \in \mathrm{rows}(P) \\ \mathbf{p}\cdot\mathbf{d} = \mathbf{p}\cdot\mathbf{e}_i = 1}} \mathbf{p}$$

$\mathbf{y}_i \cdot s = 1$ iff $G\mathbf{d}$, $G\mathbf{e}_i$ both have odd parity.

$G\mathbf{d}$ has even parity $\Rightarrow$ *all* $\mathbf{y}_i \cdot s = 0$

Let y_i form rows of a matrix M, such that $M\mathbf{s} = \mathbf{0}$

Can solve for $\mathbf{s}$! ... If M has high rank. Empirically it does!

IQP: can it be fixed?

- Attack relies on properties of QR code

IQP: can it be fixed?

- Attack relies on properties of QR code
- Could pick a different G for which this attack would not succeed?

IQP: can it be fixed?

- Attack relies on properties of QR code
- Could pick a different G for which this attack would not succeed?
- Ultimately, would like to rely on standard cryptographic assumptions…

Quantum circuits for $x^2 \bmod N$

Goal: $\mathcal{U}\,|x\rangle\,|0\rangle = |x\rangle\,|x^2 \bmod N\rangle$

Quantum circuits for $x^2 \bmod N$

Goal: $\mathcal{U}\,|x\rangle\,|0\rangle = |x\rangle\,|x^2 \bmod N\rangle$

Idea: do something really quantum: compute function in phase!

Quantum circuits for $x^2 \bmod N$

Goal: $\mathcal{U}\,|x\rangle\,|0\rangle = |x\rangle\,|x^2 \bmod N\rangle$

Idea: do something really quantum: compute function in phase!

Decompose this as

$$\mathcal{U} = (\mathbb{I} \otimes \mathrm{IQFT}_N) \cdot \tilde{\mathcal{U}} \cdot (\mathbb{I} \otimes \mathrm{QFT}_N)$$

with

$$\tilde{\mathcal{U}}\,|x\rangle\,|z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right)|x\rangle\,|z\rangle$$

Quantum circuits for $x^2 \bmod N$

Goal: $\mathcal{U}\,|x\rangle\,|0\rangle = |x\rangle\,|x^2 \bmod N\rangle$

Idea: do something really quantum: compute function in phase!

Decompose this as

$$\mathcal{U} = (\mathbb{I} \otimes \mathrm{IQFT}_N) \cdot \tilde{\mathcal{U}} \cdot (\mathbb{I} \otimes \mathrm{QFT}_N)$$

with

$$\tilde{\mathcal{U}}\,|x\rangle\,|z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right)|x\rangle\,|z\rangle$$

Advantages:

- Everything is diagonal (it's just a phase)!
- Modulo is automatic in the phase
- Simple decomposition into few-qubit gates

New goal: $\tilde{\mathcal{U}} |x\rangle |z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right) |x\rangle |z\rangle$

Decompose using "grade school" integer multiplication:

$$a \cdot b = \sum_{i,j} 2^{i+j} a_i b_j$$

Implementation

New goal: $\tilde{\mathcal{U}} |x\rangle |z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right) |x\rangle |z\rangle$

Decompose using "grade school" integer multiplication:

$$a \cdot b = \sum_{i,j} 2^{i+j} a_i b_j$$

$$x^2 z = \sum_{i,j,k} 2^{i+j+k} x_i x_j z_k$$

Implementation

New goal: $\tilde{\mathcal{U}} \left|x\right\rangle \left|z\right\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right) \left|x\right\rangle \left|z\right\rangle$

Decompose using "grade school" integer multiplication:

$$a \cdot b = \sum_{i,j} 2^{i+j} a_i b_j$$

$$x^2 z = \sum_{i,j,k} 2^{i+j+k} x_i x_j z_k$$

$$\exp\left(2\pi i \frac{x^2}{N} z\right) = \prod_{i,j,k} \exp\left(2\pi i \frac{2^{i+j+k}}{N} x_i x_j z_k\right)$$

Implementation

New goal: $\tilde{\mathcal{U}} |x\rangle |z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right) |x\rangle |z\rangle$

$$\exp\left(2\pi i \frac{x^2}{N} z\right) = \prod_{i,j,k} \exp\left(2\pi i \frac{2^{i+j+k}}{N} x_i x_j z_k\right)$$

- Binary multiplication is AND

Implementation

New goal: $\tilde{\mathcal{U}} |x\rangle |z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right) |x\rangle |z\rangle$

$$\exp\left(2\pi i \frac{x^2}{N} z\right) = \prod_{i,j,k} \exp\left(2\pi i \frac{2^{i+j+k}}{N} x_i x_j z_k\right)$$

- Binary multiplication is AND
- "Apply phase whenever $x_i = x_j = z_k = 1$"

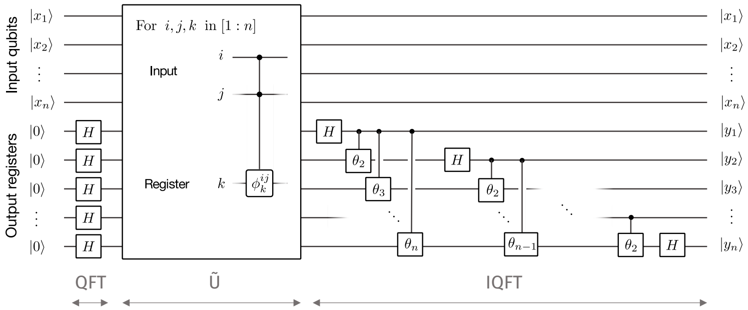
Implementation

New goal: $\tilde{\mathcal{U}} |x\rangle |z\rangle = \exp\left(2\pi i \frac{x^2}{N} z\right) |x\rangle |z\rangle$

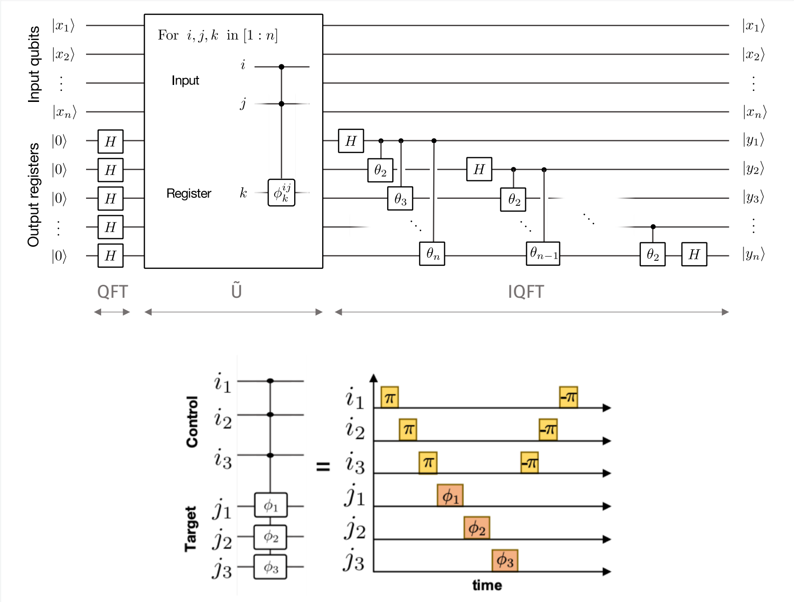
$$\exp\left(2\pi i \frac{x^2}{N} z\right) = \prod_{i,j,k} \exp\left(2\pi i \frac{2^{i+j+k}}{N} x_i x_j z_k\right)$$

- Binary multiplication is AND
- "Apply phase whenever $x_i = x_j = z_k = 1$"
- These are CCPhase gates (of arb. phase)!

Leveraging the Rydberg blockade



Leveraging the Rydberg blockade



Decisional Diffie-Hellman (DDH)

Problem (not TCF): Consider a group $\mathbb{G}$ of order N, with generator g.
Given the tuple (g, g^a, g^b, g^c), determine if $c = ab$.

Elliptic curve crypto.: $\log N \sim 160$ bits is as hard as 1024 bit factoring!!

Decisional Diffie-Hellman (DDH)

Problem (not TCF): Consider a group $\mathbb{G}$ of order N, with generator g. Given the tuple (g, g^a, g^b, g^c), determine if $c = ab$.

Elliptic curve crypto.: $\log N \sim 160$ bits is as hard as 1024 bit factoring!!

How to build a TCF?

Decisional Diffie-Hellman (DDH)

Problem (not TCF): Consider a group $\mathbb{G}$ of order N, with generator g. Given the tuple (g, g^a, g^b, g^c), determine if $c = ab$.

Elliptic curve crypto.: $\log N \sim 160$ bits is as hard as 1024 bit factoring!!

How to build a TCF?

Trapdoor [Peikert, Waters '08; Freeman et al. '10]: linear algebra in the exponent

Claw-free [GDKM et al. '21 (arXiv:2104.00687)]: collisions in linear algebra in the exponent!

Full protocol

Prover (quantum)

Verifier (classical)

Round 1

Verifier: 1. Sample $(f_i, t) \leftarrow \mathsf{Gen}(1^n)$

$\xleftarrow{f_i}$

Prover: 2. Generate state $\sum_{x=0}^{N/2} |x\rangle_{\mathsf{x}} |f_i(x)\rangle_{\mathsf{y}}$

Prover: 3. Measure y register, yielding bitstring y
State is now $(|x_0\rangle + |x_1\rangle)_{\mathsf{x}} |y\rangle_{\mathsf{y}}$;
y register can be discarded

$\xrightarrow{y}$

Verifier: 4. Using trapdoor t compute x_0 and x_1

Verifier: 5. Randomly choose to request a preimage or continue

$\xleftarrow{\text{choice}}$

If preimage requested:

Prover: 6a. Projectively measure x register, yielding x

$\xrightarrow{x}$

Verifier: 7a. If $x \in \{x_0, x_1\}$ return Accept

Otherwise, continue:

Round 2

Verifier: 6b. Choose random bitstring r

$\xleftarrow{r}$

Prover: 7b. Add one ancilla b; use CNOTs to compute
$|r \cdot x_0\rangle_{\mathsf{b}} |x_0\rangle_{\mathsf{x}} + |r \cdot x_1\rangle_{\mathsf{b}} |x_1\rangle_{\mathsf{x}}$ where
$r \cdot x$ is bitwise inner product

Prover: 8b. Measure x register in Hadamard basis, yielding a string d. Discard x, state is now
$|\psi\rangle_{\mathsf{b}} \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$

$\xrightarrow{d}$

Verifier: 9b. Using r, x_0, x_1, d, determine $|\psi\rangle_{\mathsf{b}}$

Round 3

Verifier: 10b. Choose random $m \in \{\frac{\pi}{4}, -\frac{\pi}{4}\}$

$\xleftarrow{m}$

Prover: 11b. Measure ancilla b in the rotated basis

$$\left\{ \begin{matrix} \cos\left(\frac{m}{2}\right)|0\rangle + \sin\left(\frac{m}{2}\right)|1\rangle \\ \cos\left(\frac{m}{2}\right)|1\rangle - \sin\left(\frac{m}{2}\right)|0\rangle \end{matrix} \right\},$$ yielding a bit b

$\xrightarrow{b}$

Verifier: 11b. If b was likely given $|\psi\rangle_{\mathsf{b}}$ return Accept