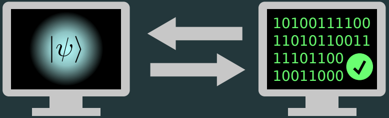
Classical verification of quantum computational advantage



Gregory D. Kahanamoku-Meyer

March 15, 2022

arXiv:2104.00687 (theory)
arXiv:2112.05156 (expt.)

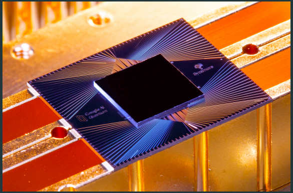
Theory collaborators:

- Norman Yao (Berkeley → Harvard)
- Umesh Vazirani (Berkeley)
- Soonwon Choi (Berkeley → MIT)

Berkeley
UNIVERSITY OF CALIFORNIA

Quantum computational advantage

Recent first experimental demonstrations:



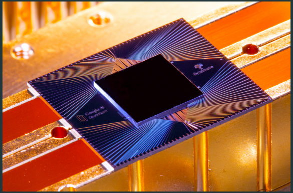
Random circuit sampling
[Arute et al., Nature '19]

Gaussian boson sampling
[Zhong et al., Science '20]

• • •

Quantum computational advantage

Recent first experimental demonstrations:



Random circuit sampling
[Arute et al., Nature '19]

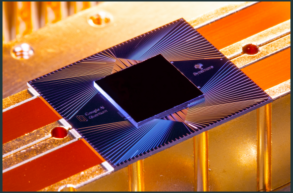
Gaussian boson sampling
[Zhong et al., Science '20]

• • •

Biggest experiments impossible to classically simulate

Quantum computational advantage

Recent first experimental demonstrations:



Random circuit sampling
[Arute et al., Nature '19]

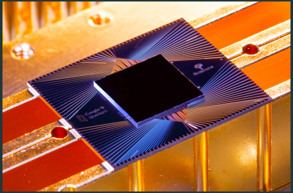
Gaussian boson sampling
[Zhong et al., Science '20]

• • •

Biggest experiments impossible to classically simulate—how do we verify the output?

Quantum computational advantage

Recent first experimental demonstrations:



Random circuit sampling
[Arute et al., Nature '19]

Gaussian boson sampling
[Zhong et al., Science '20]

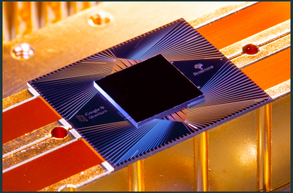
• • •

Biggest experiments impossible to classically simulate—how do we verify the output?

“[Rule] out alternative [classical] hypotheses” [Zhong et al.]

Quantum computational advantage

Recent first experimental demonstrations:



Random circuit sampling
[Arute et al., Nature '19]

Gaussian boson sampling
[Zhong et al., Science '20]

• • •

Biggest experiments impossible to classically simulate—how do we verify the output?

"[Rule] out alternative [classical] hypotheses" [Zhong et al.]

Quantum is the only reasonable explanation for observed behavior,
under some assumptions about the inner workings of the device

“Black-box” quantum computational advantage

Stronger: rule out all classical hypotheses, even pathological!

“Black-box” quantum computational advantage

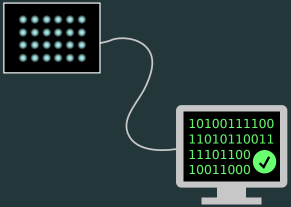
Stronger: rule out all classical hypotheses, even pathological!

Explicitly perform an efficiently-verifiable “cryptographic proof of quantum power”

“Black-box” quantum computational advantage

Stronger: rule out all classical hypotheses, even pathological!

Explicitly perform an efficiently-verifiable “cryptographic proof of quantum power”



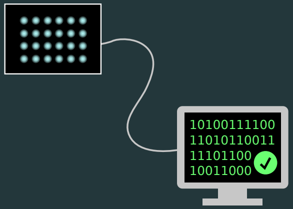
Local: robust demonstration of the power of quantum computation

"Qubits prove their power to humanity"

“Black-box” quantum computational advantage

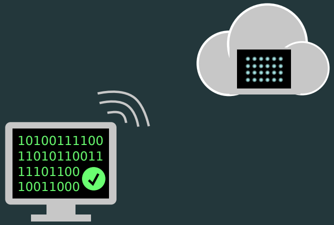
Stronger: rule out all classical hypotheses, even pathological!

Explicitly perform an efficiently-verifiable “cryptographic proof of quantum power”



Local: robust demonstration of the power of quantum computation

"Qubits prove their power to humanity"



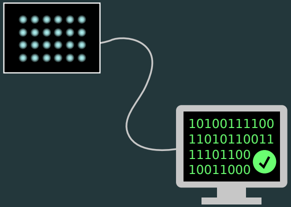
Remote: validate an untrusted quantum device over the internet

"Website proves its power to user"

“Black-box” quantum computational advantage

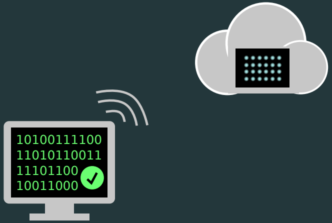
Stronger: rule out all classical hypotheses, even pathological!

Explicitly perform an efficiently-verifiable “cryptographic proof of quantum power”



Local: robust demonstration of the power of quantum computation

"Qubits prove their power to humanity"



Remote: validate an untrusted quantum device over the internet

"Website proves its power to user"

Reframing: disprove null hypothesis that output was generated classically.

Noisy intermediate scale verifiable quantum advantage

Trivial solution: Shor's algorithm

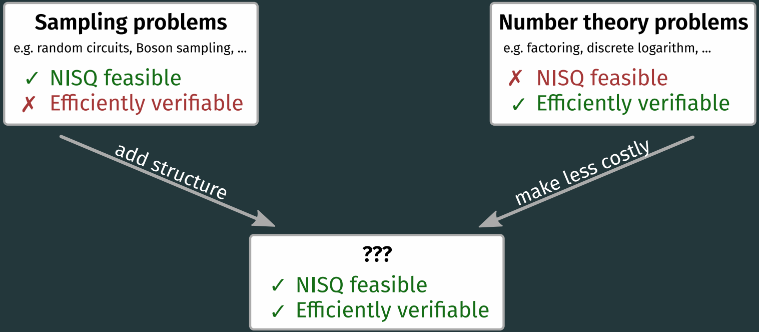
Noisy intermediate scale verifiable quantum advantage

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

Noisy intermediate scale 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

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?

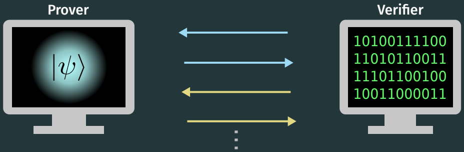
This constitutes a **zero-knowledge interactive proof**.

You (color blind) ⇔ Classical verifier
Seeing color ⇔ Quantum capability

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

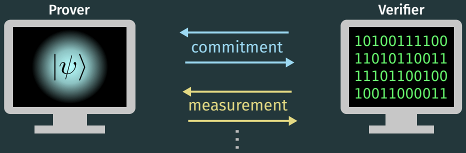
Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier



Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier

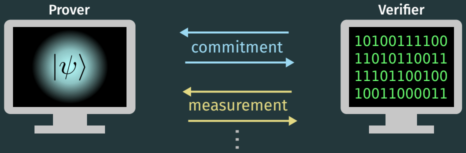


Round 1: Prover commits to holding a specific quantum state

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

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

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

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

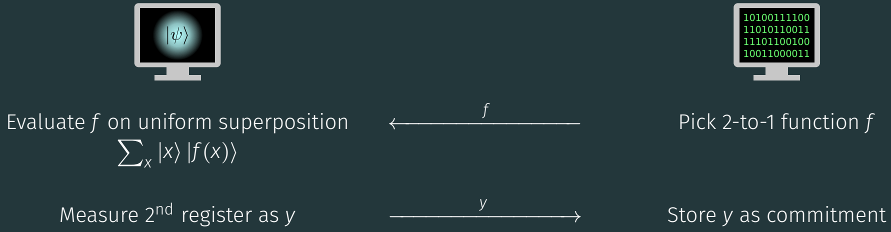
How does the prover commit to a state?

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 2nd register as y $\xrightarrow{\quad y \quad}$ Store y as commitment

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 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

$f(x) = x^2 \bmod N$, 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

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
- **Trapdoor:** Function is easily inverted with knowledge of p, q

Trapdoor claw-free function example

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

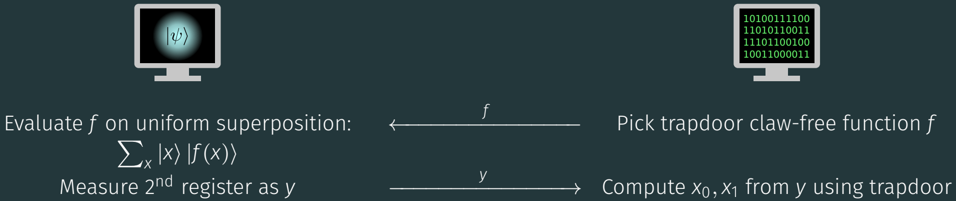
Function is actually 4-to-1 but collisions like $\{x, -x\}$ are trivial—set domain to integers in range $[0, N/2]$.

Properties:

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

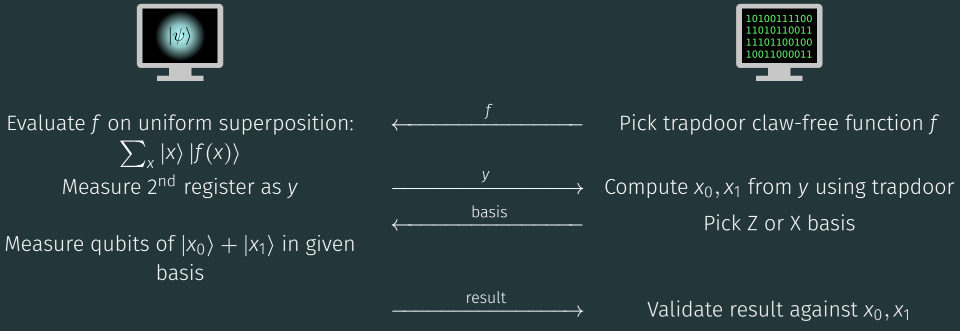


Evaluate f on uniform superposition: $\sum_x |x\rangle\,|f(x)\rangle$ $\xleftarrow{f}$ Pick trapdoor claw-free function f

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

arXiv:1804.00640. Can be extended to verify arbitrary quantum computations! arXiv:1804.01082

Brakerski, Christiano, Mahadev, Vazirani, Vidick '18



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

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

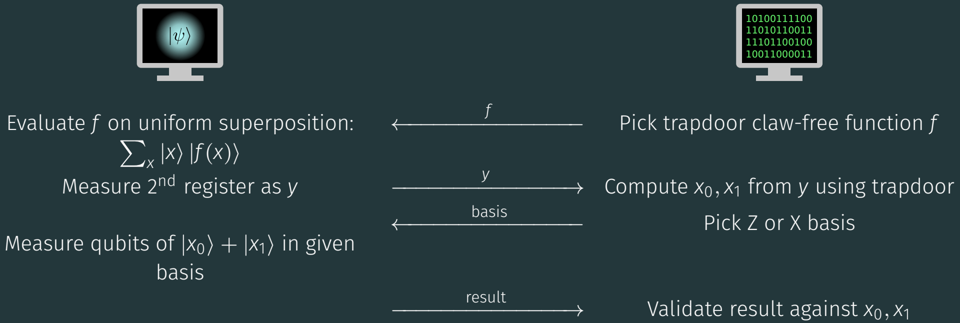
$\xleftarrow{\text{basis}}$ Pick Z or X basis

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

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

Brakerski, Christiano, Mahadev, Vazirani, Vidick ’18

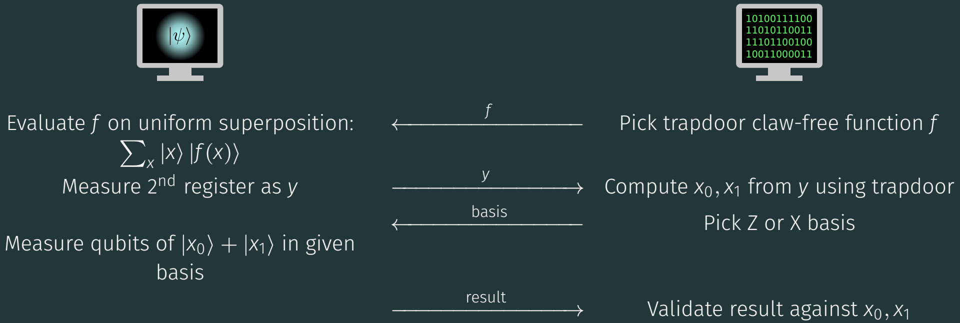
Evaluate f on uniform superposition: $\sum_x \|x\rangle \|f(x)\rangle$	$\xleftarrow{f}$	Pick trapdoor claw-free function f
Measure 2nd register as y	$\xrightarrow{y}$	Compute x_0, x_1 from y using trapdoor
	$\xleftarrow{\text{basis}}$	Pick Z or X basis
Measure qubits of $\|x_0\rangle + \|x_1\rangle$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

arXiv:1804.00640. Can be extended to verify arbitrary quantum computations! arXiv:1804.01082

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

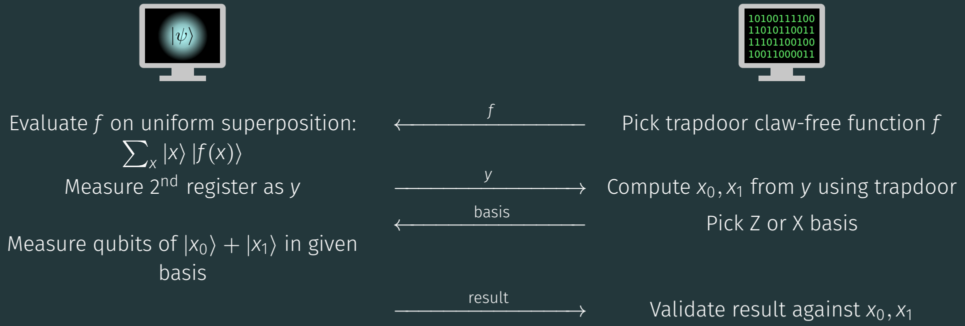
Z basis: get x_0 or x_1

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

Brakerski, Christiano, Mahadev, Vazirani, Vidick ’18



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

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

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

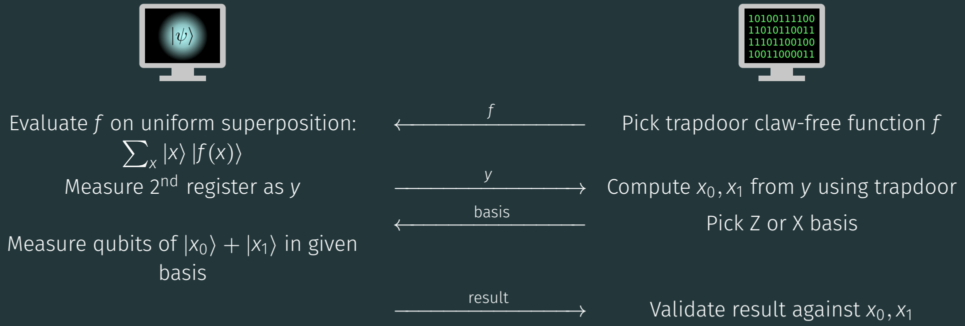
Measure qubits of $|x_0\rangle + |x_1\rangle$ in given basis

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

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

arXiv:1804.00640. Can be extended to verify arbitrary quantum computations! arXiv:1804.01082

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

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

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]

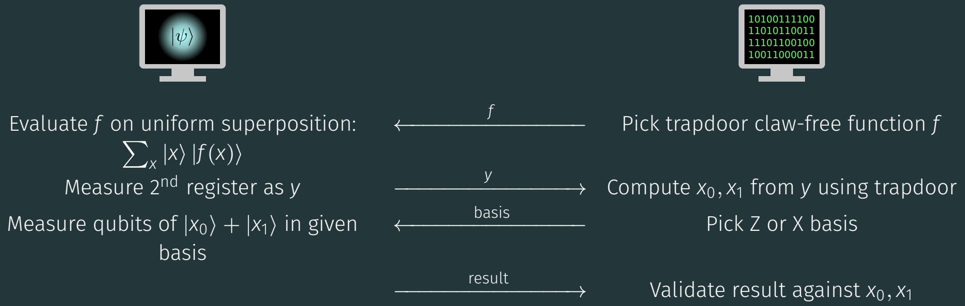
Can we do the same in the **standard model**? Yes! [3]

[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)

Brakerski, Christiano, Mahadev, Vazirani, Vidick ’18



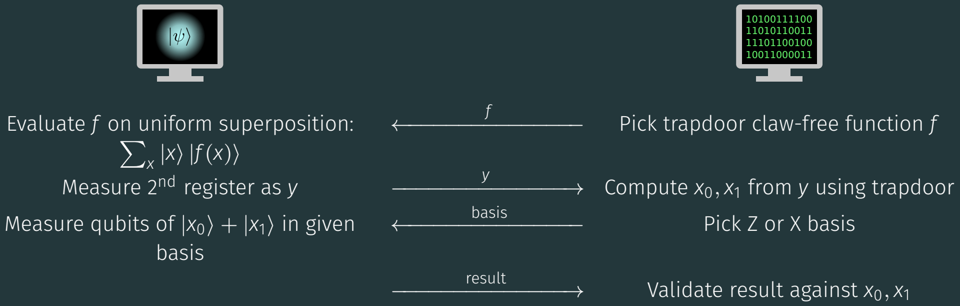
Evaluate f on uniform superposition: $\sum_x |x\rangle |f(x)\rangle$ $\xleftarrow{\quad f \quad}$ Pick trapdoor claw-free function f

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

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

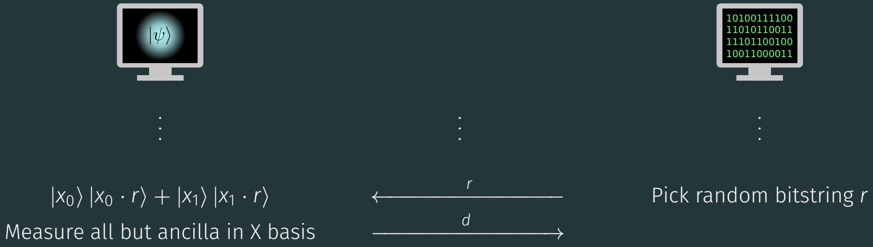
Measure qubits of $|x_0\rangle + |x_1\rangle$ in given basis $\xleftarrow{\quad \text{basis} \quad}$ Pick Z or X basis

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

Replace X basis measurement with “single-qubit Bell test”

Interactive measurement: computational Bell test

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



$|x_0\rangle |x_0 \cdot r\rangle + |x_1\rangle |x_1 \cdot r\rangle$

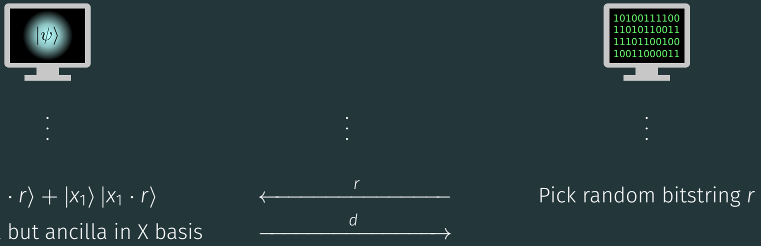
Measure all but ancilla in X basis

Pick random bitstring r

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

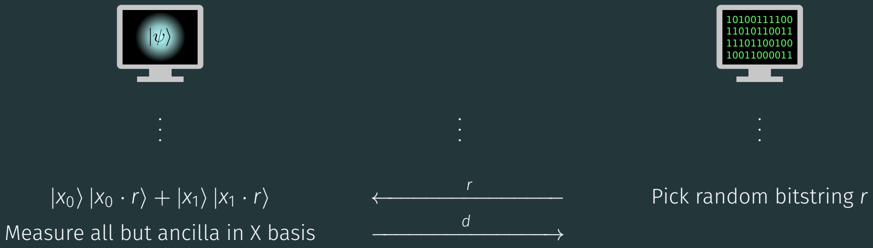
Measure all but ancilla in X basis $\xrightarrow{\quad d \quad}$

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

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.”



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

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

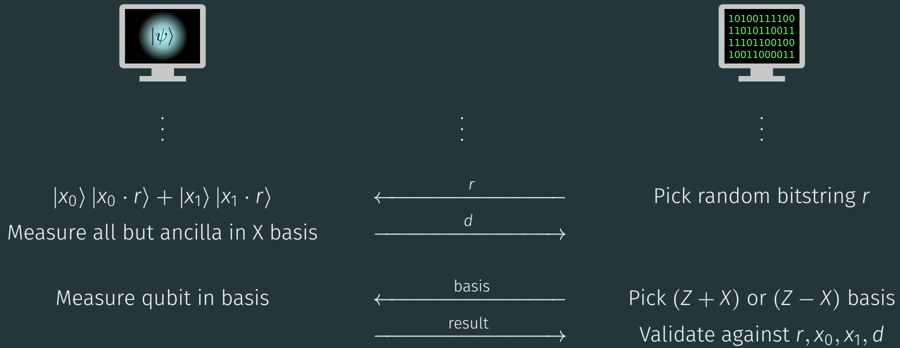
Now 1-qubit state: $|0\rangle$ or $|1\rangle$ if $x_0 \cdot r = x_1 \cdot r$, otherwise $|+\rangle$ or $|-\rangle$. Polarization hidden via:

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.”



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

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

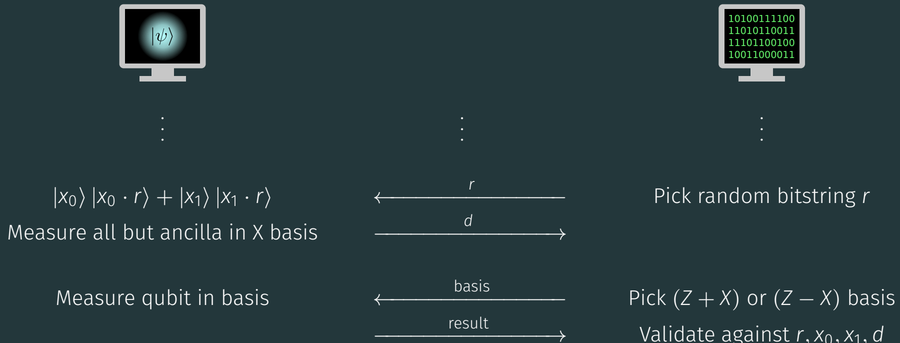
Measure qubit in basis $\xleftarrow{\text{basis}}$ Pick $(Z + X)$ or $(Z - X)$ basis

$\xrightarrow{\text{result}}$ 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.”



$|x_0\rangle\,|x_0 \cdot r\rangle + |x_1\rangle\,|x_1 \cdot r\rangle$ $\xleftarrow{\quad r \quad}$ 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: $p_Z + 4p_{\mathrm{Bell}} \lesssim 4$

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

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

Overview: efficiently verifiable quantum advantage protocol

- Existing experiments (e.g. random circuits) not verifiable at scale
- 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
- 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
- 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
- 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 measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland (→ Duke)

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

Dr. Daiwei Zhu

Prof. Crystal Noel

Prof. Christopher Monroe

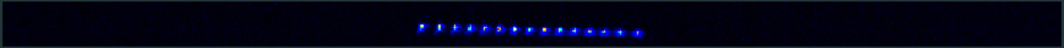
and others!

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland (→ Duke)

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

Partial measurement:

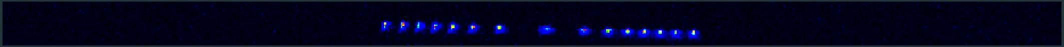


Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland (→ 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 (→ 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 (→ 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 (→ 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 (→ 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 (→ 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 (→ Duke)

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

Partial measurement:

Intermediate measurements in the lab

TIQI

Trapped Ion Quantum Information lab at U. Maryland (→ Duke)

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

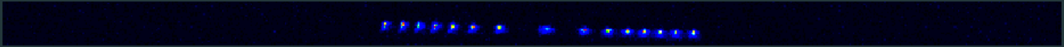
Partial measurement:

Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland (→ Duke)

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

Partial measurement:

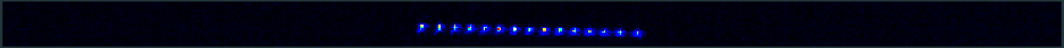


Intermediate measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland (→ Duke)

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

Partial measurement:

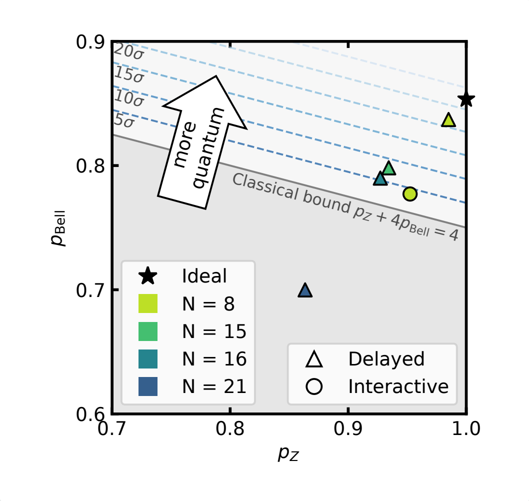


Interactive proofs on a few qubits

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

Looking forward

Bottleneck: Evaluating TCF on quantum superposition

Improving implementation of the protocol:

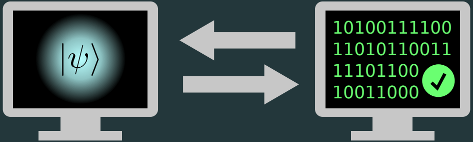
- Preliminary implementation of $x^2 \bmod N$ at scale has depth 10^5—optimize it!
- Co-design circuits for specific hardware (Rydberg implementation in paper)
- $x^2 \bmod N$ requires at minimum 500-1000 qubits for classical hardness—search for new claw-free functions?

Improving the protocol itself:

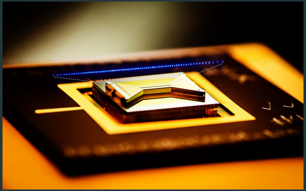
- Remove trapdoor—symmetric key/hash-based cryptography
- Explore other protocols (verifiable sampling?)

Questions?

arXiv:2104.00687 (theory)



arXiv:2112.05156 (experiment)



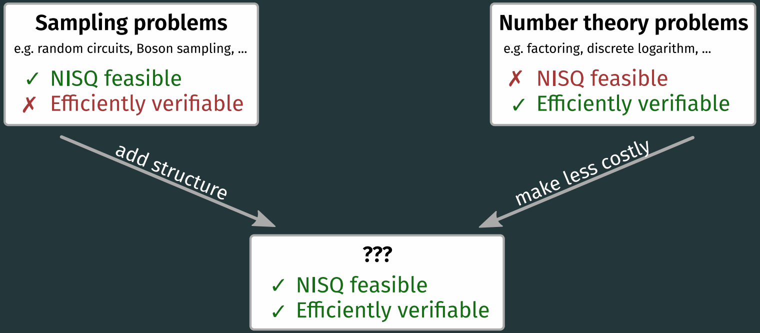
Gregory D. Kahanamoku-Meyer

gregdmeyer.github.io

Backup!

Noisy intermediate scale 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

Adding structure to sampling problems

Generically: seems hard.

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

Adding structure to sampling problems

Generically: seems hard.

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

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

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

Adding structure to sampling problems

Generically: seems hard.

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

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]: Can hide a secret in H, such that evolving and sampling gives results correlated with secret

Adding structure to sampling problems

Generically: seems hard.

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

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]: Can 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

Generically: seems hard.

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

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]: Can 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

[GDKM 2019]: Classical algorithm to extract the secret from H

Adding structure to sampling problems

Generically: seems hard.

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

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

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

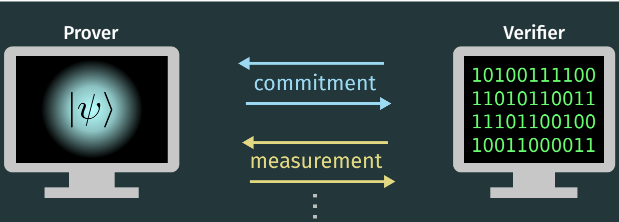
[Shepherd, Bremner 2009]: Can hide a secret in H, such that evolving and sampling gives results correlated with secret

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

[GDKM 2019]: Classical algorithm to extract the secret from H

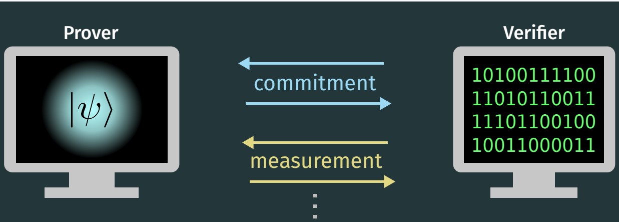
Adding structure opens opportunities for classical cheating

Hardness proof: rewinding



From a “proof of hardness” perspective:

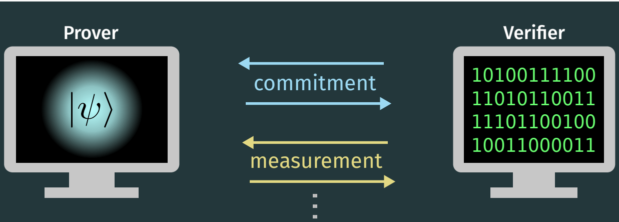
Hardness proof: rewinding



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

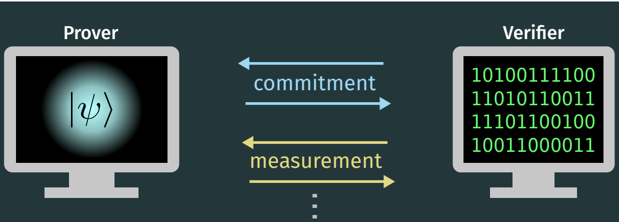
Hardness proof: rewinding



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



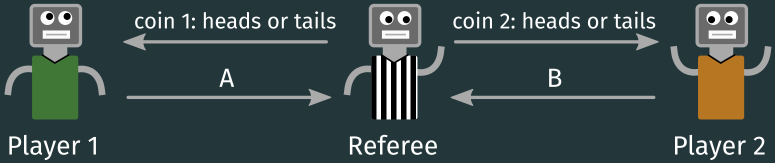
From a “proof of hardness” perspective:

- **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!

The CHSH game (Bell test)

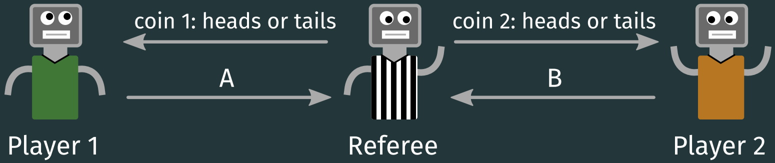
Cooperative two-player game; players can't communicate (non-local).



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

The CHSH game (Bell test)

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



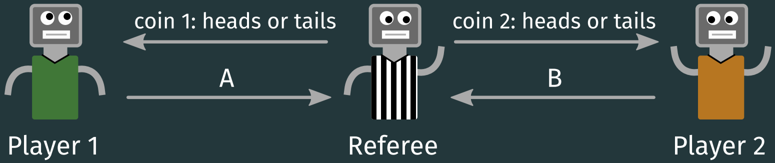
If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

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

Can we do better with entanglement?

The CHSH game (Bell test)

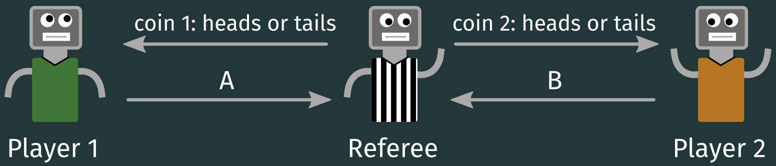
Cooperative two-player game; players can't communicate (non-local).



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

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

The CHSH game (Bell test)



coin 1: heads or tails

A

coin 2: heads or tails

B

Player 1

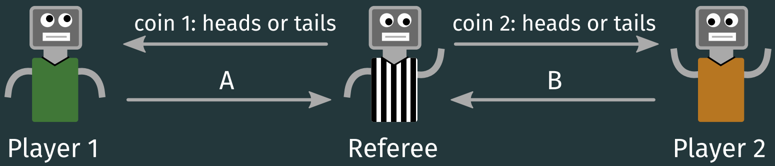
Referee

Player 2

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

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

The CHSH game (Bell test)

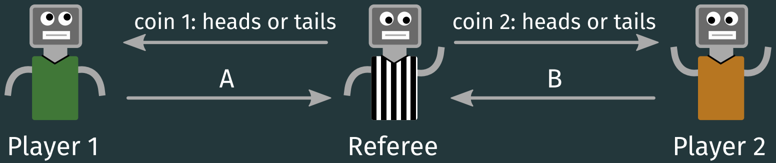


If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

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!

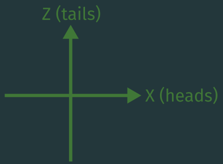
The CHSH game (Bell test)



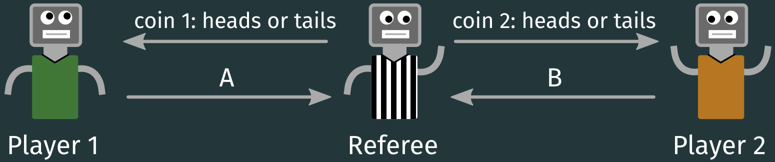
If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

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

Aligned basis → same result; antialigned → opposite result!



The CHSH game (Bell test)



coin 1: heads or tails

A

coin 2: heads or tails

B

Player 1

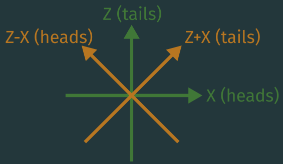
Referee

Player 2

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 → same result; antialigned → opposite result!



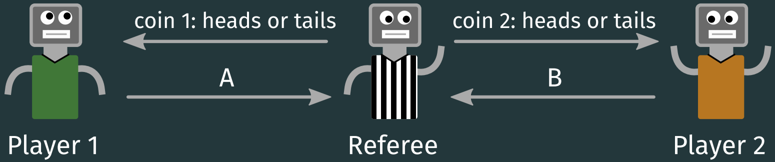
Z (tails)

Z-X (heads)

Z+X (tails)

X (heads)

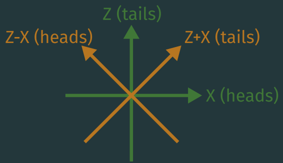
The CHSH game (Bell test)



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 → same result; antialigned → opposite result!



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

Classical: 75%

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

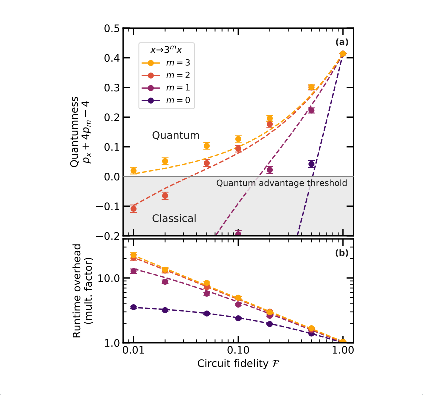
How to deal with high fidelity requirement? Naively need $\sim 83\%$ overall circuit fidelity to pass.

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

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

Getting rid of strong claw-free property helps!

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

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.

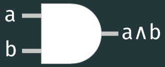
Protocol allows us to make circuits irreversible!

Technique: taking out the garbage

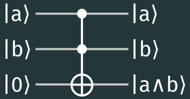
Goal: $\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity



Classical AND



Quantum AND (Toffoli)

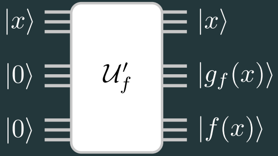
Technique: taking out the garbage

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 $g_f(x)$:



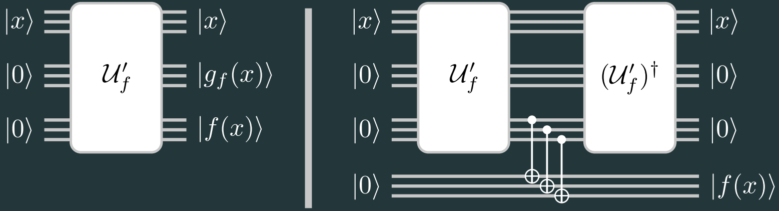
Technique: taking out the garbage

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 $g_f(x)$:



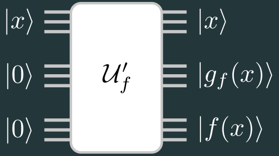
Technique: taking out the garbage

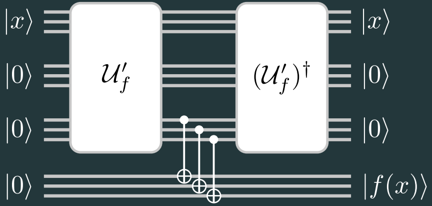
Goal: $\mathcal{U}_f |x\rangle |0^{\otimes n}\rangle = |x\rangle |f(x)\rangle$

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!

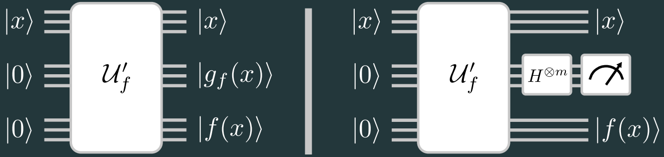
Technique: taking out the garbage

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?)

Quantum circuits for $x^2 \bmod N$

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

Quantum circuits for $x^2 \bmod N$

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

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

Quantum circuits for $x^2 \bmod N$

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

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

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

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:

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

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

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

- Binary multiplication is AND

Implementation

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

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

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

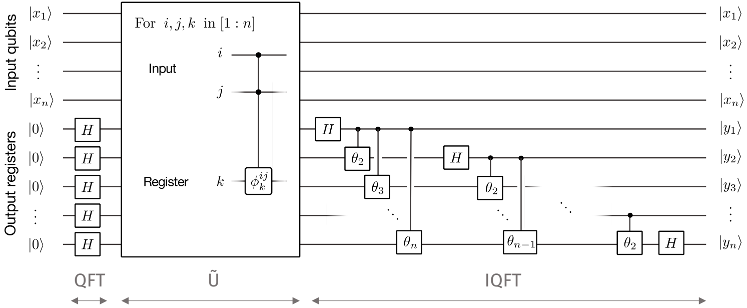
Implementation

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

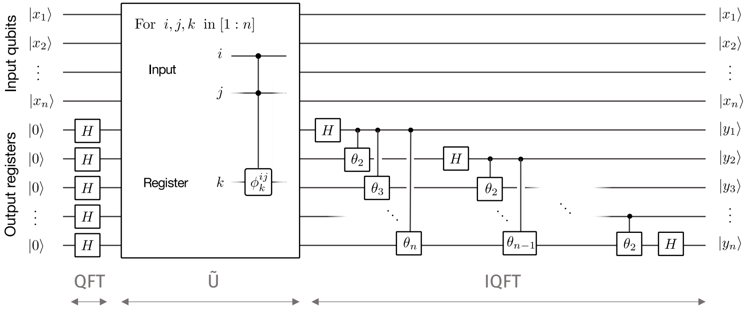
$$\exp\left(2\pi i \frac{x^2}{N} z\right) = \prod_{i,j,k} \exp\left(2\pi i \frac{2^{i+j+k}}{N} x_i x_j z_k\right)$$

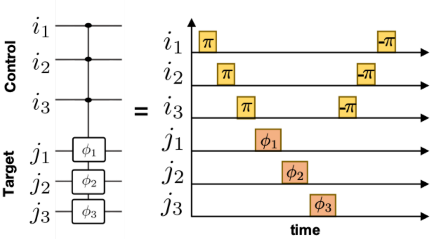
- Binary multiplication is AND
- “Apply phase whenever $x_i = x_j = z_k = 1$”
- These are CCPhase gates (of arb. phase)!

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 $c = ab$.

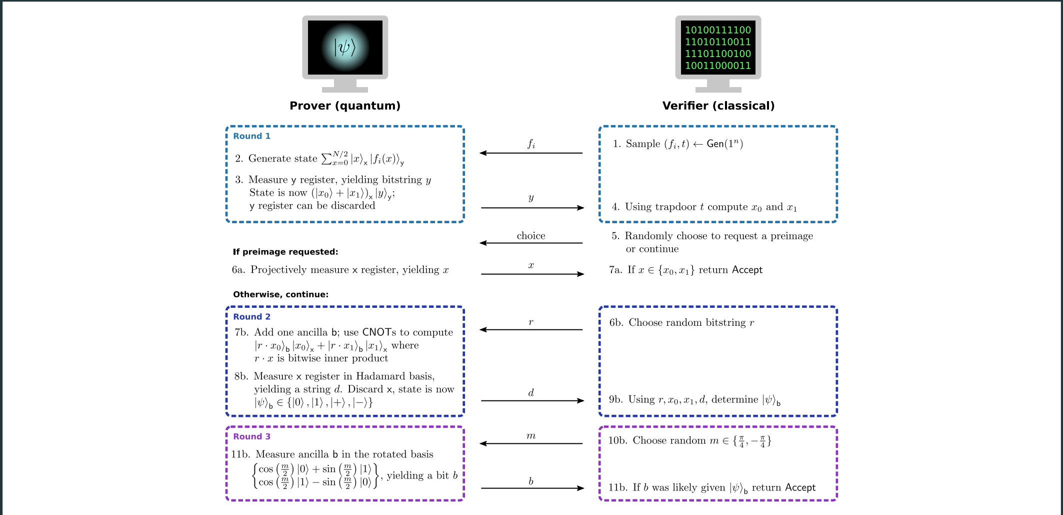
Elliptic curve crypto.: $\log N \sim 160$ bits is as hard as 1024 bit factoring!!

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

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

$\xleftarrow{f_i}$

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

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

$\xrightarrow{y}$

4. Using trapdoor t compute x_0 and x_1

5. Randomly choose to request a preimage or continue

$\xleftarrow{\text{choice}}$

If preimage requested:

6a. Projectively measure x register, yielding x

$\xrightarrow{x}$

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

Otherwise, continue:

Round 2

6b. Choose random bitstring r

$\xleftarrow{r}$

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

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

$\xrightarrow{d}$

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

Round 3

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

$\xleftarrow{m}$

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

$\xrightarrow{b}$

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