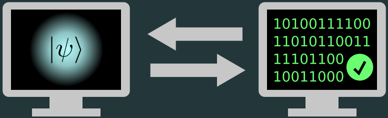


Classical verification of quantum computational advantage

Gregory D. Kahanamoku-Meyer

July 14, 2021

arXiv:2104.00687

Theory collaborators:

Norman Yao (UCB)
Umesh Vazirani (UCB)
Soonwon Choi (UCB -> MIT)

SIMONS INSTITUTE for the Theory of Computing

Berkeley UNIVERSITY OF CALIFORNIA

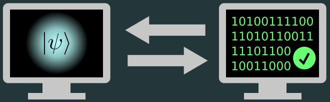
“Black-box” proofs of quantumness

Efficiently-verifiable test that only quantum computers can pass.

“Black-box” proofs of quantumness

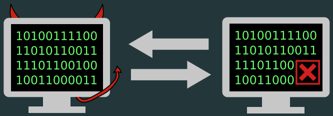
Efficiently-verifiable test that only quantum computers can pass.

For polynomially-bounded classical verifier:



Completeness

$\exists$ BQP prover s.t. Verifier accepts w.p. $> 2/3$



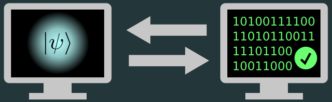
Soundness

$\forall$ BPP provers, Verifier accepts w.p. $< 1/3$

“Black-box” proofs of quantumness

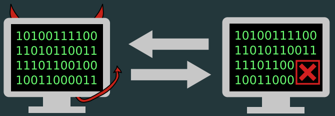
Efficiently-verifiable test that only quantum computers can pass.

For polynomially-bounded classical verifier:



Completeness

$\exists$ BQP prover s.t. Verifier accepts w.p. $> 2/3$



Soundness

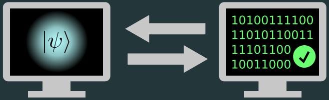
$\forall$ BPP provers, Verifier accepts w.p. $< 1/3$

Fully classical verifier (and comms.),

“Black-box” proofs of quantumness

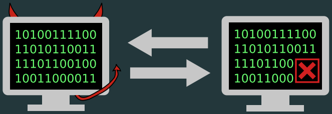
Efficiently-verifiable test that only quantum computers can pass.

For polynomially-bounded classical verifier:



Completeness

$\exists$ BQP prover s.t. Verifier accepts w.p. $> 2/3$



Soundness

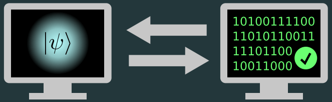
$\forall$ BPP provers, Verifier accepts w.p. $< 1/3$

Fully classical verifier (and comms.), single black-box prover.

“Black-box” proofs of quantumness

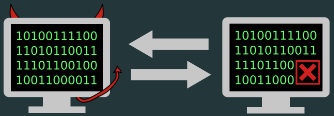
Efficiently-verifiable test that only quantum computers can pass.

For polynomially-bounded classical verifier:



Completeness

$\exists$ BQP prover s.t. Verifier accepts w.p. $> 2/3$



Soundness

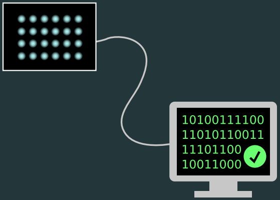
$\forall$ BPP provers, Verifier accepts w.p. $< 1/3$

Fully classical verifier (and comms.), single black-box prover.

Disprove null hypothesis that prover is classical!

“Black-box” proofs of quantumness

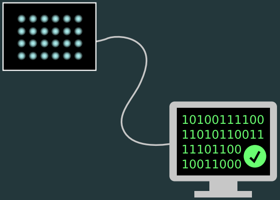
Efficiently-verifiable test that only quantum computers can pass.



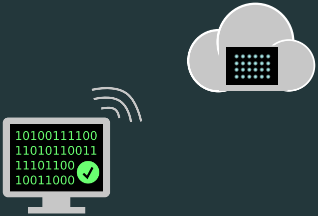
Local: powerfully refute the extended Church-Turing thesis

“Black-box” proofs of quantumness

Efficiently-verifiable test that only quantum computers can pass.



Local: powerfully refute the extended Church-Turing thesis



Remote: validate an untrusted quantum cloud service

NISQ verifiable quantum advantage

Trivial solution: Shor's algorithm

NISQ verifiable quantum advantage

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

NISQ 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!

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)

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

[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

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 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

NISQ 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

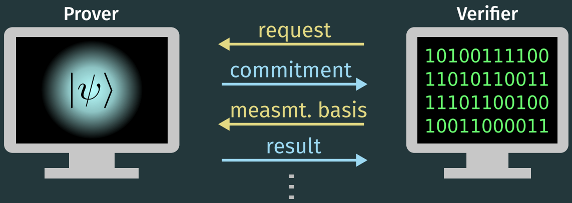
add structure

make less costly

???

- ✓ NISQ feasible
- ✓ Efficiently verifiable

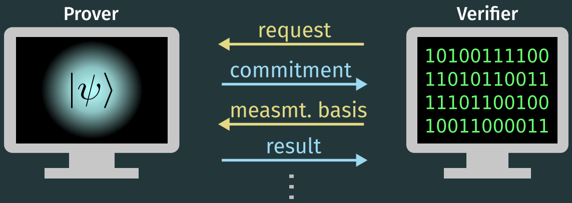
Interactive proofs of quantumness



Round 1: Prover commits to a specific quantum state

Round 2+: Verifier asks for measurement in specific basis

Interactive proofs of quantumness



Round 1: Prover commits to a specific quantum state

Round 2+: Verifier asks for measurement in specific basis

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 collision-resistant (claw-free) function f.

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

How does the prover commit to a state?

Consider a **2-to-1** collision-resistant (claw-free) function f.

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

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

LWE protocol

Prover		Verifier
		
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

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

LWE protocol

Prover		Verifier
		
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 standard or Hadamard basis
Measure qubits of $|x_0\rangle + |x_1\rangle$ in given basis	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

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

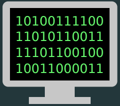
LWE protocol

Prover		Verifier
		
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 standard or Hadamard basis
Measure qubits of $\|x_0\rangle + \|x_1\rangle$ in given basis		
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Subtlety: claw-free does *not* imply hardness of generating measurement outcomes!

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

LWE protocol

Prover		Verifier
		
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 standard or Hadamard basis
Measure qubits of $\|x_0\rangle + \|x_1\rangle$ in given basis		
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Subtlety: claw-free does *not* imply hardness of generating measurement outcomes!
Learning-with-Errors TCF has adaptive hardcore bit

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

Trapdoor claw-free functions

TCF	Trapdoor	Claw-free	Adaptive hard-core bit
LWE [1]	✓	✓	✓
$x^2 \bmod N$ [3]	✓	✓	✗
Ring-LWE [2]	✓	✓	✗
Diffie-Hellman [3]	✓	✓	✗

[1] Brakerski, Christiano, Mahadev, Vazirani, Vidick '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

TCF	Trapdoor	Claw-free	Adaptive hard-core bit
LWE [1]	✓	✓	✓
$x^2 \bmod N$ [3]	✓	✓	✗
Ring-LWE [2]	✓	✓	✗
Diffie-Hellman [3]	✓	✓	✗

BKVV ’20 [2]: Non-interactive protocol without adaptive hardcore bit, in random oracle model

[1] Brakerski, Christiano, Mahadev, Vazirani, Vidick ’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

TCF	Trapdoor	Claw-free	Adaptive hard-core bit
LWE [1]	✓	✓	✓
$x^2 \bmod N$ [3]	✓	✓	✗
Ring-LWE [2]	✓	✓	✗
Diffie-Hellman [3]	✓	✓	✗

BKVV '20 [2]: Non-interactive protocol without adaptive hardcore bit, in random oracle model

Can we remove AHCB in the standard model?

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

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

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

LWE protocol

Prover		Verifier
		
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 standard or Hadamard basis
	$\xrightarrow{\text{result}}$	Validate result against x_0, x_1

Replace Hadamard basis measurement with “1-player CHSH”

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

Interactive measurement: computational Bell test

Replace Hadamard basis measurement with two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

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

$\|x_0\rangle \|x_0 \cdot r\rangle + \|x_1\rangle \|x_1 \cdot r\rangle$	$\xleftarrow{\quad r \quad}$	Pick random bitstring r
Measure all but ancilla in Hadamard basis	$\xrightarrow{\quad d \quad}$	

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

Interactive measurement: computational Bell test

Replace Hadamard basis measurement with two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

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

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

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

Now single-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

Replace Hadamard basis measurement with two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

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

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

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

Now single-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

Replace Hadamard basis measurement with two-step process: "condense" x_0, x_1 into a single qubit, and then do a "Bell test."

$\vdots$ $\quad$ $\vdots$ $\quad$ $\vdots$

$\lvert x_0\rangle \lvert x_0 \cdot r\rangle + \lvert x_1\rangle \lvert x_1 \cdot r\rangle$	$\xleftarrow{\quad r \quad}$	Pick random bitstring r
Measure all but ancilla in Hadamard 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)

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_s: Success rate for standard basis measurement.

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

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_s: Success rate for standard basis measurement.

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

Under assumption of claw-free function:

Classical bound: $p_s + 4p_{\text{CHSH}} - 4 < \text{negl}(n)$

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_s: Success rate for standard basis measurement.

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

Under assumption of claw-free function:

Classical bound: $p_s + 4p_{\text{CHSH}} - 4 < \text{negl}(n)$

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

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_s: Success rate for standard basis measurement.

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

Under assumption of claw-free function:

Classical bound: $p_s + 4p_{\text{CHSH}} - 4 < \text{negl}(n)$

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

$p_s + 4p_{\text{CHSH}} - 4 = \sqrt{2} - 1 \approx \mathbf{0.414}$

Computational Bell test: classical bound

Run protocol many times, collect statistics.

p_s: Success rate for standard basis measurement.

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

Under assumption of claw-free function:

Classical bound: $p_s + 4p_{\text{CHSH}} - 4 < \text{negl}(n)$

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

$p_s + 4p_{\text{CHSH}} - 4 = \sqrt{2} - 1 \approx \mathbf{0.414}$

Note: Let $p_s = 1$. Then for p_{CHSH}:

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

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

Challenges for implementation

- Partial measurement

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction

Challenges for implementation

- Partial measurement
    - Required for multi-round classical interaction
- Fidelity requirement

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction
- Fidelity requirement
  - High fidelity needed to pass classical bound

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction
- Fidelity requirement
  - High fidelity needed to pass classical bound
- Circuit sizes

Challenges for implementation

- Partial measurement
    - Required for multi-round classical interaction
- Fidelity requirement
    - High fidelity needed to pass classical bound
- Circuit sizes
    - Need to implement public-key crypto. on a superposition

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Prof. Christopher Munroe

Dr. Daiwei Zhu

Dr. Crystal Noel

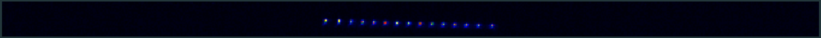
and others!

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

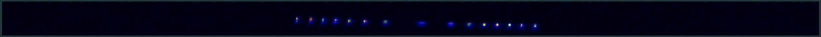


Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:



Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

Partial measurements in the lab

TIQI

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

Partial measurements in the lab

TIQI

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

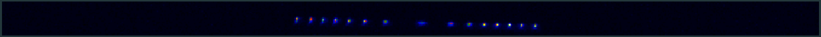
Partial measurement:

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

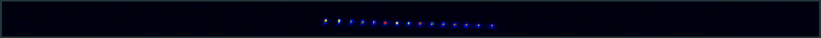
Partial measurement:



Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:



Technique: postselection

How to deal with high fidelity requirement? Need $\sim 83\%$ fidelity in general to pass.

Technique: postselection

How to deal with high fidelity requirement? Need $\sim 83\%$ fidelity in general to pass.

Can show: a prover holding $(|x_0\rangle + |x_1\rangle)\,|y\rangle$ with ϵ phase coherence passes!

Technique: postselection

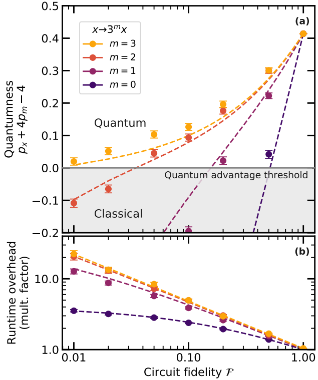
How to deal with high fidelity requirement? Need $\sim 83\%$ fidelity in general to pass.

Can show: 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? Need $\sim 83\%$ fidelity in general 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 adaptive hardcore bit helps!

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

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 adaptive hardcore bit 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.

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 adaptive hardcore bit 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.

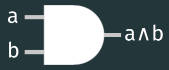
Protocol allows us to make circuits irreversible!

Technique: taking out the garbage

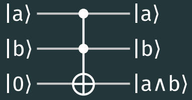
Goal: $\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity



Classical AND



Quantum AND (Toffoli)

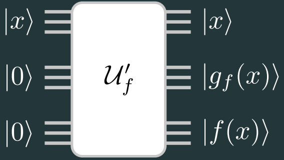
Technique: taking out the garbage

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

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity

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



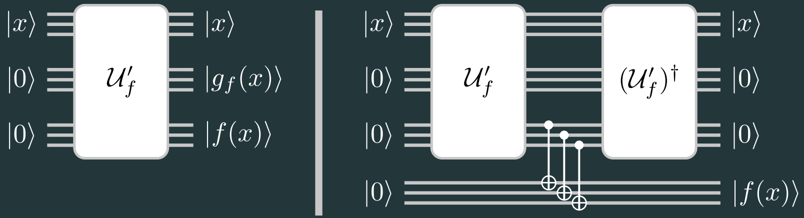
Technique: taking out the garbage

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 $g_f(x)$:



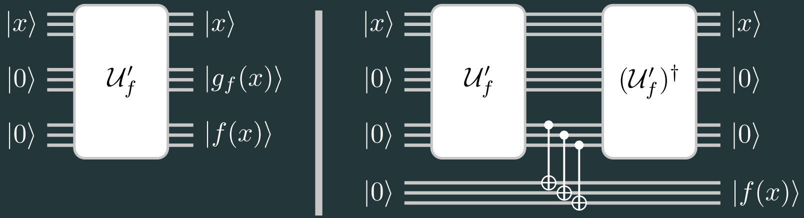
Technique: taking out the garbage

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 $g_f(x)$:



Lots of time and space overhead!

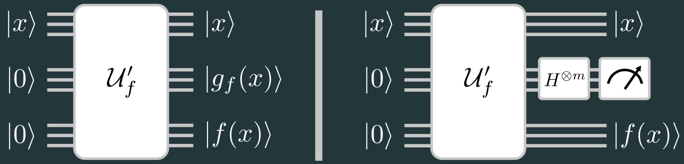
Technique: taking out the garbage

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

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity

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



Can we “measure them away” instead?

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in Hadamard 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 Hadamard 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 Hadamard 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 Hadamard 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!

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in Hadamard 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!

Technique: taking out the garbage

Measure garbage bits $g_f(x)$ in Hadamard 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 $x^2 \mod N$ costs only 10^6 Toffoli gates.

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

“In the box” ideas (not necessarily bad):

- Find more efficient TCFs

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

"In the box" ideas (not necessarily bad):

- Find more efficient TCFs
- Better quantum circuits for TCFs

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

"In the box" ideas (not necessarily bad):

- Find more efficient TCFs
- Better quantum circuits for TCFs
- ... public-key cryptography is just slow

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

"In the box" ideas (not necessarily bad):

- Find more efficient TCFs
- Better quantum circuits for TCFs
- … public-key cryptography is just slow

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

"In the box" ideas (not necessarily bad):

- Find more efficient TCFs
- Better quantum circuits for TCFs
- … public-key cryptography is just slow

"Box-adjacent" ideas:

- Explore other protocols (fix IQP and make it fast?)

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

"In the box" ideas (not necessarily bad):

- Find more efficient TCFs
- Better quantum circuits for TCFs
- ... public-key cryptography is just slow

"Box-adjacent" ideas:

- Explore other protocols (fix IQP and make it fast?)
- Symmetric key/hash-based cryptography?

Bottleneck: Evaluating TCF on quantum superposition

"In the box" ideas (not necessarily bad):

- Find more efficient TCFs
- Better quantum circuits for TCFs
- … public-key cryptography is just slow

"Box-adjacent" ideas:

- Explore other protocols (fix IQP and make it fast?)
- Symmetric key/hash-based cryptography?

Paths forward

Bottleneck: Evaluating TCF on quantum superposition

"In the box" ideas (not necessarily bad):

- Find more efficient TCFs
- Better quantum circuits for TCFs
- … public-key cryptography is just slow

"Box-adjacent" ideas:

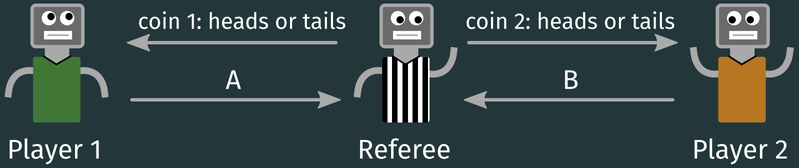
- Explore other protocols (fix IQP and make it fast?)
- Symmetric key/hash-based cryptography?

Way outside the box?

Backup!

The CHSH game (Bell test)

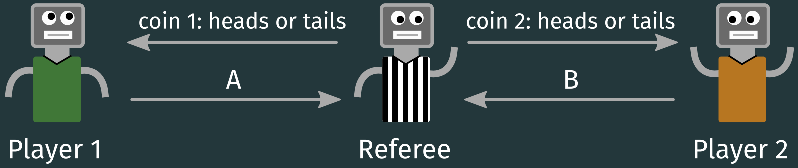
Two-player cooperative game.



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

The CHSH game (Bell test)

Two-player cooperative game.

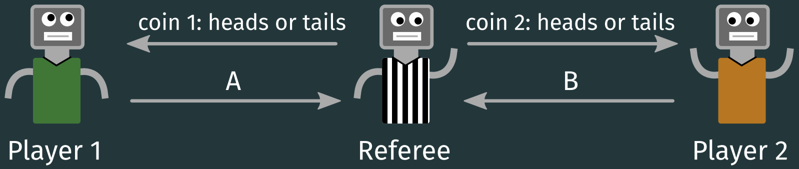


If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Two players sharing a Bell pair:

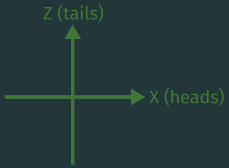
The CHSH game (Bell test)

Two-player cooperative game.



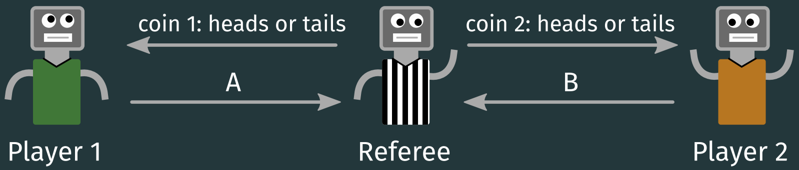
If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Two players sharing a Bell pair:



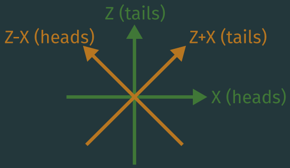
The CHSH game (Bell test)

Two-player cooperative game.



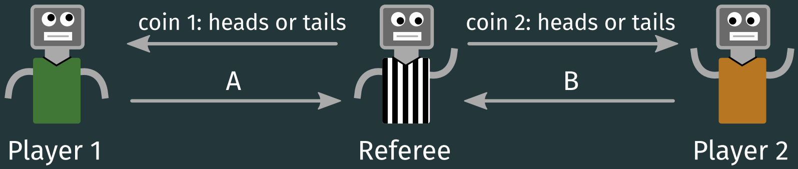
If anyone receives tails, want $A = B$. If both get heads, want $A \neq B$.

Two players sharing a Bell pair:



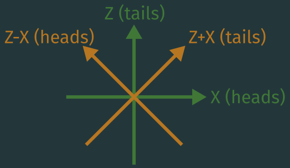
The CHSH game (Bell test)

Two-player cooperative game.



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

Two players sharing a Bell pair:



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

Classical: 75%

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)

Prover (quantum)	Message	Verifier (classical)
Round 1	$\xleftarrow{f_i}$	1. Sample $(f_i, t) \leftarrow \mathsf{Gen}(1^n)$
2. Generate state $\sum_{x=0}^{N/2} \lvert x\rangle_{\mathsf{x}} \lvert f_i(x)\rangle_{\mathsf{y}}$		
3. Measure y register, yielding bitstring y State is now $(\lvert x_0\rangle + \lvert x_1\rangle)_{\mathsf{x}} \lvert y\rangle_{\mathsf{y}}$; y register can be discarded	$\xrightarrow{y}$	4. Using trapdoor t compute x_0 and x_1
	$\xleftarrow{\text{choice}}$	5. Randomly choose to request a preimage or continue
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	$\xleftarrow{r}$	6b. Choose random bitstring r
7b. Add one ancilla b; use CNOTs to compute $\lvert r \cdot x_0\rangle_{\mathsf{b}} \lvert x_0\rangle_{\mathsf{x}} + \lvert r \cdot x_1\rangle_{\mathsf{b}} \lvert x_1\rangle_{\mathsf{x}}$ where $r \cdot x$ is bitwise inner product		
8b. Measure x register in Hadamard basis, yielding a string d. Discard x, state is now $\lvert\psi\rangle_{\mathsf{b}} \in \{\lvert 0\rangle, \lvert 1\rangle, \lvert +\rangle, \lvert -\rangle\}$	$\xrightarrow{d}$	9b. Using r, x_0, x_1, d, determine $\lvert\psi\rangle_{\mathsf{b}}$
Round 3	$\xleftarrow{m}$	10b. Choose random $m \in \{\frac{\pi}{4}, -\frac{\pi}{4}\}$
11b. Measure ancilla b in the rotated basis $\left\{\begin{matrix} \cos\left(\frac{m}{2}\right)\lvert 0\rangle + \sin\left(\frac{m}{2}\right)\lvert 1\rangle \\ \cos\left(\frac{m}{2}\right)\lvert 1\rangle - \sin\left(\frac{m}{2}\right)\lvert 0\rangle \end{matrix}\right\}$, yielding a bit b	$\xrightarrow{b}$	11b. If b was likely given $\lvert\psi\rangle_{\mathsf{b}}$ return Accept