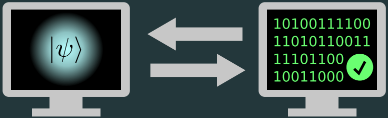


Classical verification of quantum computational advantage

Gregory D. Kahanamoku-Meyer

October 8, 2021

arXiv:1912.05547
arXiv:2104.00687

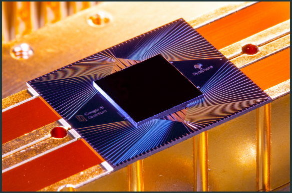
Theory collaborators:

Norman Yao (Berkeley Physics)
Umesh Vazirani (Berkeley CS)
Soonwon Choi (MIT Physics)

Berkeley
UNIVERSITY OF CALIFORNIA

Quantum computational advantage

Recent experimental demonstrations:

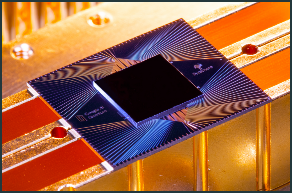


Random circuit sampling
[Arute et al., Nature '19]

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

Quantum computational advantage

Recent experimental demonstrations:



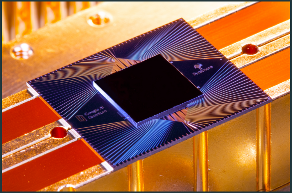
Random circuit sampling
[Arute et al., Nature '19]

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

Largest experiments → impossible to classically simulate

Quantum computational advantage

Recent experimental demonstrations:



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

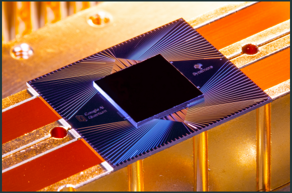
Gaussian boson sampling
[Zhong et al., Science '20]

Largest experiments → impossible to classically simulate

> "... [Rule] out alternative [classical] hypotheses that might be plausible in this experiment" [Zhong et al.]

Quantum computational advantage

Recent experimental demonstrations:



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

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

Largest experiments → impossible to classically simulate

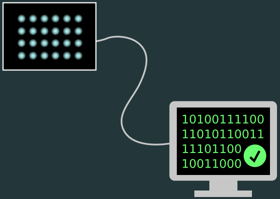
> "… [Rule] out alternative [classical] hypotheses that might be plausible in this experiment" [Zhong et al.]
> Quantum is the only reasonable explanation for observed behavior

“Black-box” proofs of quantumness

Stronger: rule out all classical hypotheses, even adversarial!

“Black-box” proofs of quantumness

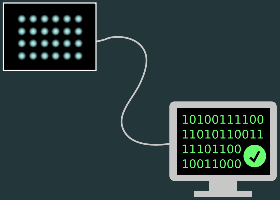
Stronger: rule out all classical hypotheses, even adversarial!



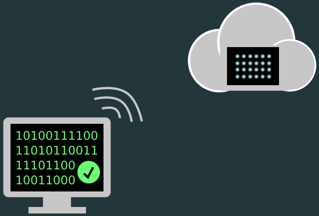
Local: powerfully refute the extended Church-Turing thesis

“Black-box” proofs of quantumness

Stronger: rule out all classical hypotheses, even adversarial!



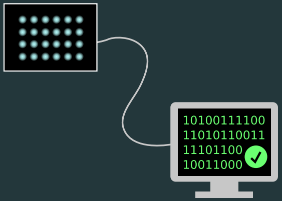
Local: powerfully refute the extended Church-Turing thesis



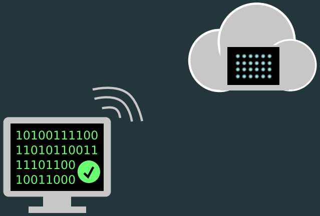
Remote: validate an untrusted quantum cloud service

“Black-box” proofs of quantumness

Stronger: rule out all classical hypotheses, even adversarial!



Local: powerfully refute the extended Church-Turing thesis



Remote: validate an untrusted quantum cloud service

Proof not specific to quantum mechanics: disprove null hypothesis that output was generated classically.

Need computational assumption—really an “argument”

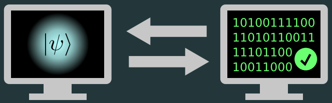
“Black-box” proofs of quantumness

Efficiently-verifiable test that only quantum computers can pass.

“Black-box” proofs of quantumness

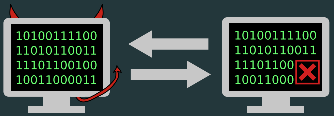
Efficiently-verifiable test that only quantum computers can pass.

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. $< 1/3$

NISQ verifiable quantum advantage

Trivial solution: integer factorization

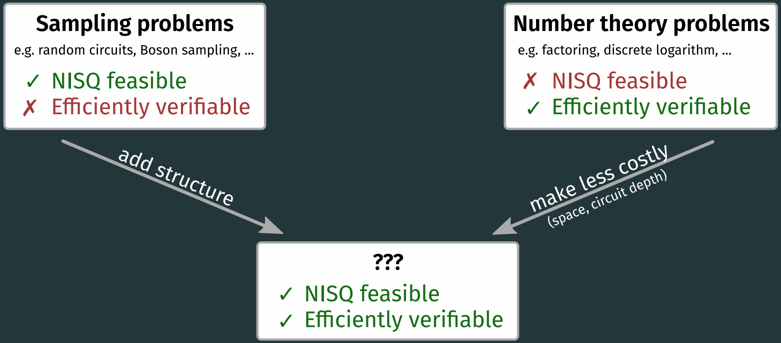
NISQ verifiable quantum advantage

Trivial solution: integer factorization... but we want near-term!

NISQ verifiable quantum advantage

Trivial solution: integer factorization... but we want 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
(space, circuit depth)

???

✓ NISQ feasible

✓ Efficiently verifiable

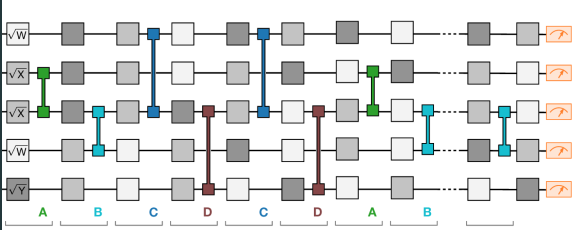
Sampling problems

Task: generate samples from a “hard” probability distribution.

Sampling problems

Task: generate samples from a “hard” probability distribution.

Random circuit sampling:



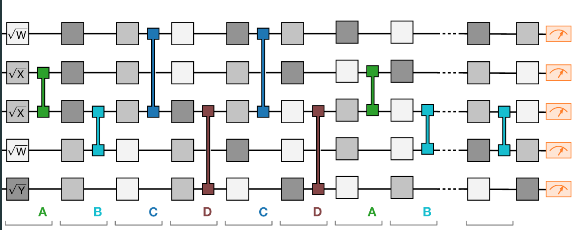
Arute et al. 2019

- Specify distribution via a quantum circuit

Sampling problems

Task: generate samples from a “hard” probability distribution.

Random circuit sampling:



Arute et al. 2019

- Specify distribution via a quantum circuit
- Intuitive classical hardness: no structure → need to simulate quantum, which is hard

Adding structure to sampling problems

Idea: some *property* of samples that we can check?

Adding structure to sampling problems

Idea: some *property* of samples that we can check?

Generically: seems difficult to make work.

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

Adding structure to sampling problems

Idea: some *property* of samples that we can check?

Generically: seems difficult to make work.

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

IQP circuits [Shepherd and Bremner, '08]:

- Hide a secret string $\mathbf{s}$ in the quantum circuit
- Set up circuit so it is *biased* to generate samples $\mathbf{x}$ with $\mathbf{x}^\intercal \cdot \mathbf{s} = 0$.

IQP circuits [Shepherd and Bremner, ’08]

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

IQP circuits [Shepherd and Bremner, ’08]

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

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

Example:

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

IQP circuits [Shepherd and Bremner, '08]

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

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

Example:

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

Distribution of sampling result X:

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

IQP circuits [Shepherd and Bremner, ’08]

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

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

Example:

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

Distribution of sampling result X:

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

Bremner, Jozsa, Shepherd ’11: classically sampling IQP circuits would collapse polynomial heirarchy

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

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

Let $\theta = \pi/8$ and P have the form:

$$P = \left[\begin{array}{c} G \\ \hline R \end{array}\right]$$

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

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

Let $\theta = \pi/8$ and P have the form:

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

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

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

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

Let $\theta = \pi/8$ and P have the form:

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

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

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

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

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

Let $\theta = \pi/8$ and P have the form:

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

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

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

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

IQP: Hiding s

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

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

IQP: Hiding s

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

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

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

permute rows, Gauss-Jordan columns

$$P'\mathbf{s}' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Scrambling preserves quantum success rate.

IQP: Hiding s

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

$$P = \left[\begin{array}{c} G \\ \hline R \end{array}\right] \qquad P\boldsymbol{s} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \xrightarrow{\text{permute rows, Gauss-Jordan columns}} \quad P'\boldsymbol{s}' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Scrambling preserves quantum success rate.

Conjecture [SB ’08]: Scrambling P cryptographically hides G (and equivalently s)

IQP: Classical strategy

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

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

IQP: Classical strategy

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

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

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

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

IQP: Classical strategy

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

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

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

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

Then:

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

IQP: Classical strategy

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

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

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

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

Then:

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

IQP: Classical strategy

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

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

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

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

Then:

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

IQP: Classical strategy

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

Best classical: $\Pr[Y^\intercal \cdot s = 0] \overset{?}{=} 0.5$

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

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

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

Then:

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

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

IQP: Classical strategy [SB '08]

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

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

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

IQP: Classical strategy [SB ’08]

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

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

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

Then:

IQP: Classical strategy [SB ’08]

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

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

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

Then:

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

IQP: Classical strategy [SB '08]

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

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

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

Then:

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

IQP: Classical strategy [SB '08]

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

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

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

Then:

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

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

IQP: Classical strategy [SB ’08]

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

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

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

Then:

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

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

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

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

Key: Correlate samples to attack the key s

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$G\boldsymbol{d}$ has even parity $\Rightarrow$ *all* $\boldsymbol{y}_i \cdot s = 0$
Let y_i form rows of a matrix M, such that $M\boldsymbol{s} = \mathbf{0}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

IQP: can it be fixed?

- Attack relies on properties of QR code

IQP: can it be fixed?

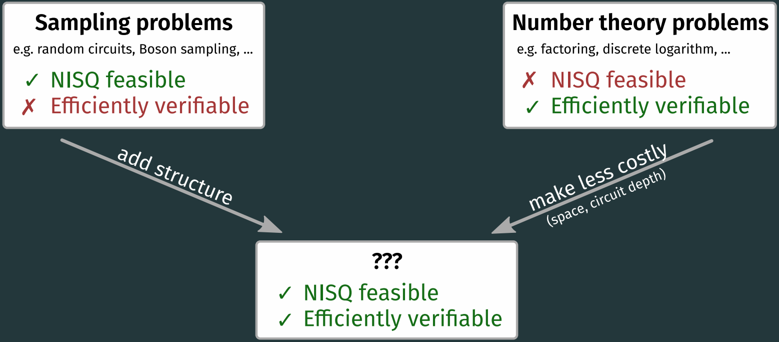
- Attack relies on properties of QR code
- Could pick a different G for which this attack would not succeed?

IQP: can it be fixed?

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

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

(space, circuit depth)

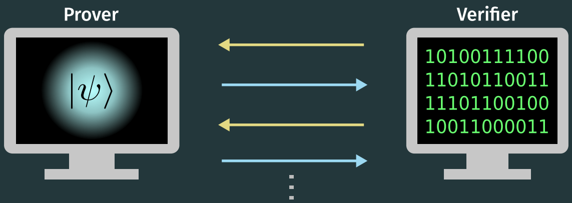
???

✓ NISQ feasible

✓ Efficiently verifiable

Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier

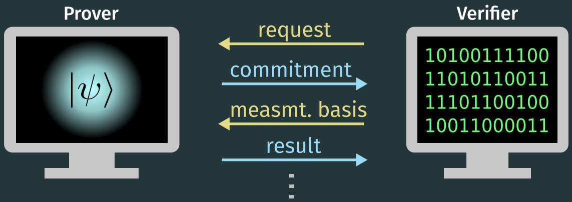


Round 1: Prover commits to a specific quantum state

Round 2+: Verifier asks for measurement in specific basis

Interactive proofs of quantumness

Multiple rounds of interaction between the prover and verifier



Round 1: Prover commits to a specific quantum state

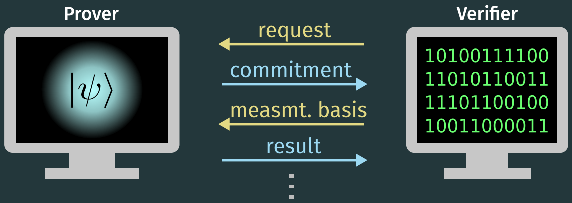
Round 2+: Verifier asks for measurement in specific basis

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).

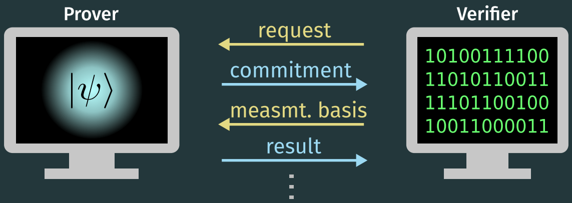
Can be extended to verify arbitrary quantum computations! (arXiv:1804.01082)

Interactive proofs of quantumness



From a proof of security perspective:

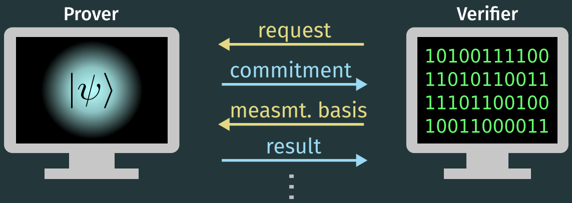
Interactive proofs of quantumness



From a proof of security perspective:

- **Classical** cheater can be rewound: extract measurement results in all choices of basis

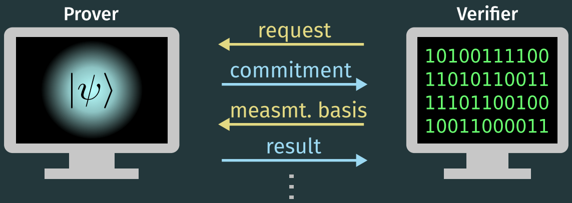
Interactive proofs of quantumness



From a proof of security perspective:

- **Classical** cheater can be rewound: extract measurement results in all choices of basis
- **Quantum** prover's measurements are irreversible

Interactive proofs of quantumness



From a proof of security perspective:

- **Classical** cheater can be rewound: 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 use post-quantum cryptography!

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

How does the prover commit to a state?

Consider a trapdoor claw-free function family (TCF) (Gen, $\{(f_i, T_i)\}$).

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

How does the prover commit to a state?

Consider a trapdoor claw-free function family (TCF) (Gen, $\{(f_i, T_i)\}$).

Prover		**Verifier**
Evaluate f_i on uniform superposition $\sum_x \lvert x\rangle \lvert f_i(x)\rangle$	$\xleftarrow{\quad f_i \quad}$	$(f_i, t_i) \leftarrow \mathrm{Gen}(1^\lambda)$
Measure 2^{nd} register as y	$\xrightarrow{\quad y \quad}$	Store y as commitment compute $(x_0, x_1) \leftarrow T_i(y, t_i)$

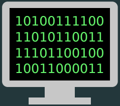
State commitment (round 1): trapdoor claw-free functions

How does the prover commit to a state?

Consider a trapdoor claw-free function family (TCF) (Gen, $\{(f_i, T_i)\}$).

Prover

Verifier

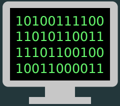


Evaluate f_i on uniform superposition $\sum_x |x\rangle |f_i(x)\rangle$ $\xleftarrow{\quad f_i \quad}$ $(f_i, t_i) \leftarrow \text{Gen}(1^\lambda)$

Measure 2nd register as y $\xrightarrow{\quad y \quad}$ Store y as commitment compute $(x_0, x_1) \leftarrow T_i(y, t_i)$

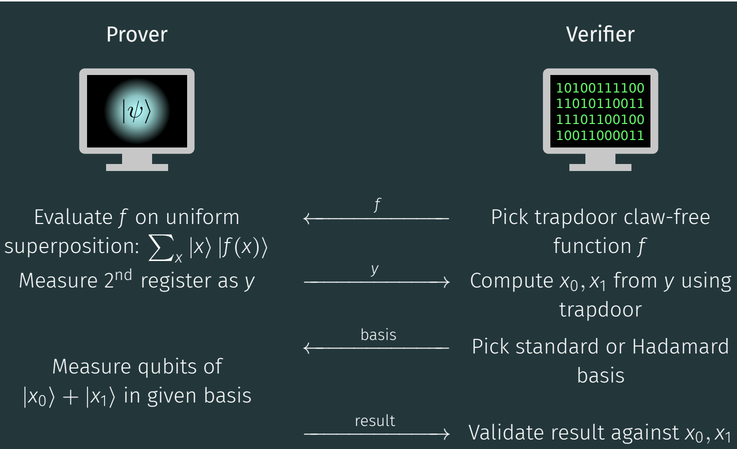
Prover has committed to the state $(|x_0\rangle + |x_1\rangle) |y\rangle$

BCMVV ’18 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)

BCMVV ’18 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

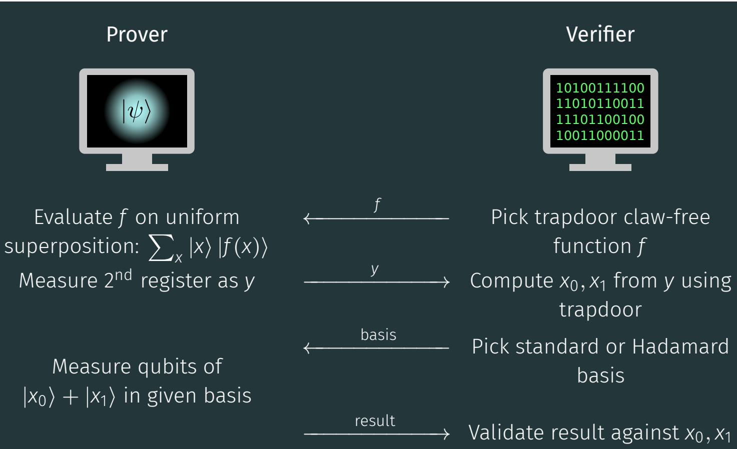
$\xleftarrow{\quad \text{basis} \quad}$ Pick standard or Hadamard basis

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

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

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

BCMVV '18 protocol



Prover | **Verifier**

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 standard or Hadamard basis

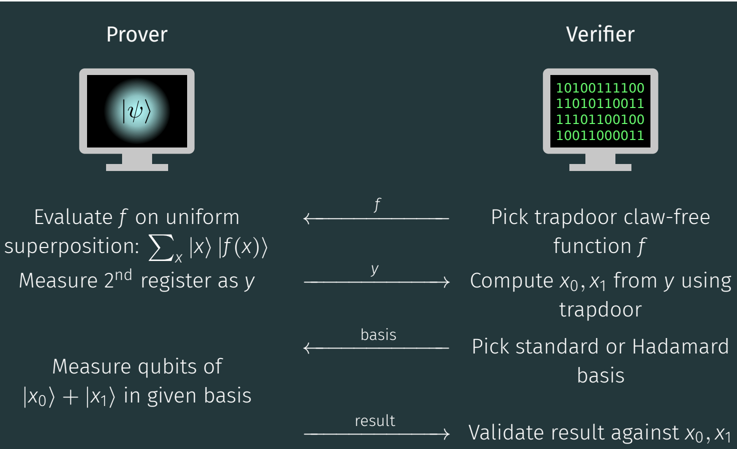
Measure qubits of $|x_0\rangle + |x_1\rangle$ in given basis

$\xrightarrow{\text{result}}$ 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)

BCMVV ’18 protocol



Prover | **Verifier**

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 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)

BCMVV ’18 protocol: adaptive hardcore bit

State after commitment round: $(|x_0\rangle + |x_1\rangle)\,|y\rangle$

BCMVV '18 protocol: adaptive hardcore bit

State after commitment round: $(|x_0\rangle + |x_1\rangle)\,|y\rangle$

Measurement outcomes:

- **Standard basis:** x_0 or x_1

BCMVV '18 protocol: adaptive hardcore bit

State after commitment round: $(|x_0\rangle + |x_1\rangle)\,|y\rangle$

Measurement outcomes:

- **Standard basis:** x_0 or x_1
- **Hadamard basis:** Some string d with $d \cdot (x_0 \oplus x_1) = 0$

BCMVV ’18 protocol: adaptive hardcore bit

State after commitment round: $(|x_0\rangle + |x_1\rangle) |y\rangle$

Measurement outcomes:

- **Standard basis:** x_0 or x_1
- **Hadamard basis:** Some string d with $d \cdot (x_0 \oplus x_1) = 0$

BCMVV ’18 protocol: adaptive hardcore bit

State after commitment round: $(|x_0\rangle + |x_1\rangle)\,|y\rangle$

Measurement outcomes:

- **Standard basis:** x_0 or x_1
- **Hadamard basis:** Some string d with $d \cdot (x_0 \oplus x_1) = 0$

Adaptive hardcore bit:

Computationally hard to generate a tuple (y, x_0, d, b) such that:

$$d \cdot (x_0 + x_1) = b$$

$$f_i(x_0) = f_i(x_1) = y$$

BCMVV ’18 protocol: adaptive hardcore bit

State after commitment round: $(|x_0\rangle + |x_1\rangle)\,|y\rangle$

Measurement outcomes:

- **Standard basis:** x_0 or x_1
- **Hadamard basis:** Some string d with $d \cdot (x_0 \oplus x_1) = 0$

Adaptive hardcore bit:

Computationally hard to generate a tuple (y, x_0, d, b) such that:

$$d \cdot (x_0 + x_1) = b$$
$$f_i(x_0) = f_i(x_1) = y$$

Note: AHCB can be post-quantum secure and protocol still works!

Trapdoor claw-free functions

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

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

$$d \cdot (x_0 \oplus x_1) = H(x_0) \oplus H(x_1)$$

[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]	✓	✓	✓
Ring-LWE [2]	✓	✓	✗
$x^2 \bmod N$ [3]	✓	✓	✗
DDH [3]	✓	✓	✗

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

$$d \cdot (x_0 \oplus x_1) = H(x_0) \oplus H(x_1)$$

Can we remove AHCB in the standard model?

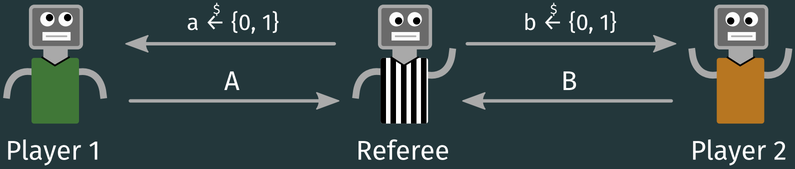
[1] Brakerski, Christiano, Mahadev, Vazirani, Vidick '18 (arXiv:1804.00640)

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

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

The CHSH game (Bell test)

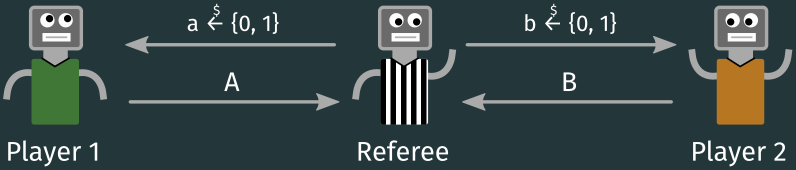
Two-player cooperative game.


$a \overset{\$}{\leftarrow} \{0, 1\}$
$b \overset{\$}{\leftarrow} \{0, 1\}$
A
B
Player 1
Referee
Player 2

Players win if $A \oplus B = a \cdot b$

The CHSH game (Bell test)

Two-player cooperative game.

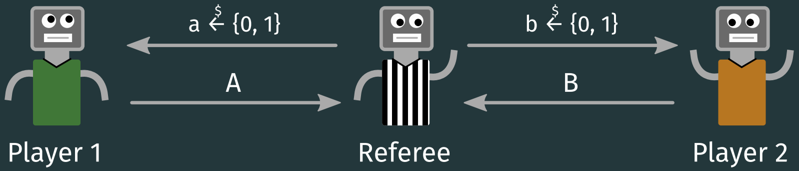

$a \overset{\$}{\leftarrow} \{0,1\}$
$b \overset{\$}{\leftarrow} \{0,1\}$
A
B
Player 1
Referee
Player 2

Players win if $A \oplus B = a \cdot b$

Classical optimal strategy: return equal values, hope $a \cdot b = 0$.
75% success rate.

The CHSH game (Bell test)

Two-player cooperative game.

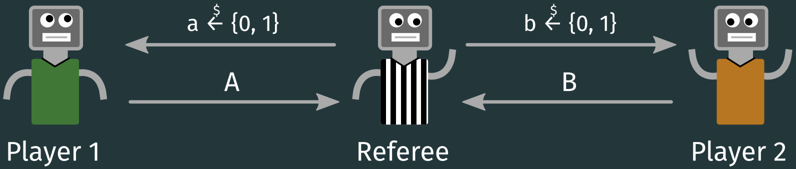

$a \overset{\$}{\leftarrow} \{0, 1\}$
$b \overset{\$}{\leftarrow} \{0, 1\}$
A
B
Player 1
Referee
Player 2

Players win if $A \oplus B = a \cdot b$

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

The CHSH game (Bell test)

Two-player cooperative game.



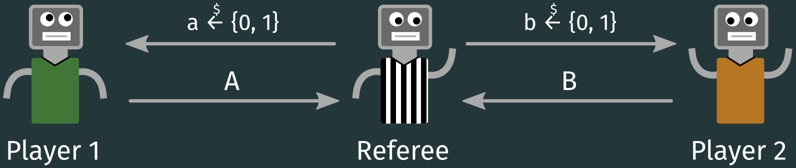
$a \overset{\$}{\leftarrow} \{0, 1\}$
$b \overset{\$}{\leftarrow} \{0, 1\}$
A
B
Player 1
Referee
Player 2

Players win if $A \oplus B = a \cdot b$

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

The CHSH game (Bell test)

Two-player cooperative game.



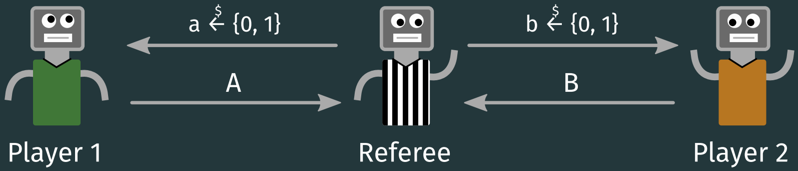
Players win if $A \oplus B = a \cdot b$

Quantum: 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)

Two-player cooperative game.



$a \overset{\$}{\leftarrow} \{0, 1\}$

A

$b \overset{\$}{\leftarrow} \{0, 1\}$

B

Player 1

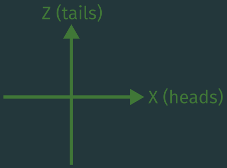
Referee

Player 2

Players win if $A \oplus B = a \cdot b$

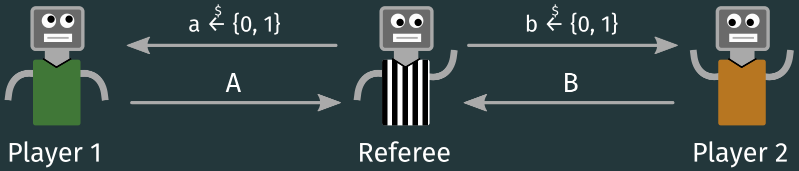
Quantum: Bell pair: $|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle = |\leftarrow\leftarrow\rangle + |\rightarrow\rightarrow\rangle = \cdots$

Aligned basis → same result;   antialigned → opposite result!



The CHSH game (Bell test)

Two-player cooperative game.



$a \overset{\$}{\leftarrow} \{0, 1\}$

A

$b \overset{\$}{\leftarrow} \{0, 1\}$

B

Player 1

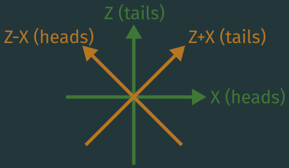
Referee

Player 2

Players win if $A \oplus B = a \cdot b$

Quantum: 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)

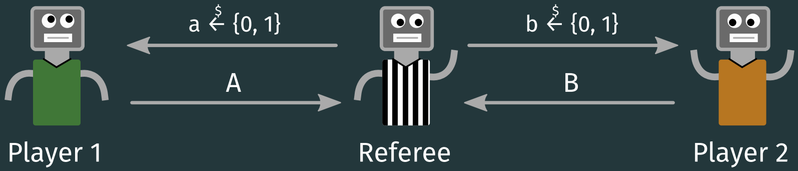
Z-X (heads)

Z+X (tails)

X (heads)

The CHSH game (Bell test)

Two-player cooperative game.



$a \xleftarrow{\$} \{0, 1\}$

A

$b \xleftarrow{\$} \{0, 1\}$

B

Player 1

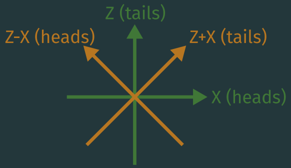
Referee

Player 2

Players win if $A \oplus B = a \cdot b$

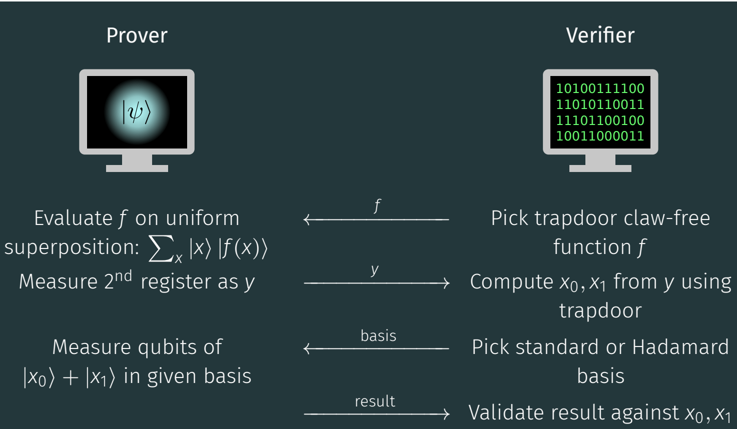
Quantum: Bell pair: $|\psi\rangle = |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle = |\leftarrow\leftarrow\rangle + |\rightarrow\rightarrow\rangle = \cdots$

Aligned basis → same result; antialigned → opposite result!



Quantum: $\cos^2(\pi/8) \approx 85\%$
Classical: 75%

BCMVV ’18 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

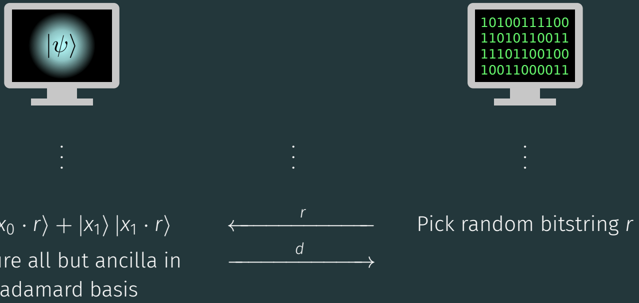
$\xrightarrow{\text{result}}$ Validate result against x_0, x_1

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."



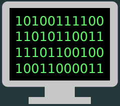
$|x_0\rangle |x_0 \cdot r\rangle + |x_1\rangle |x_1 \cdot r\rangle$ $\xleftarrow{r}$ Pick random bitstring r

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

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.”



⋮ ⋮ ⋮

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

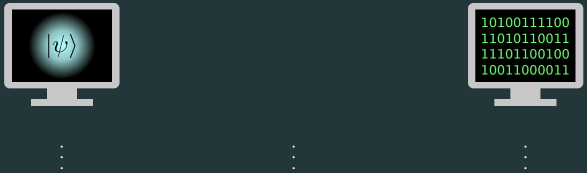
Measure all but ancilla in Hadamard basis $\xrightarrow{\quad d \quad}$

Single-qubit state: $|x_0 \cdot r\rangle + |x_1 \cdot r\rangle$

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.”



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

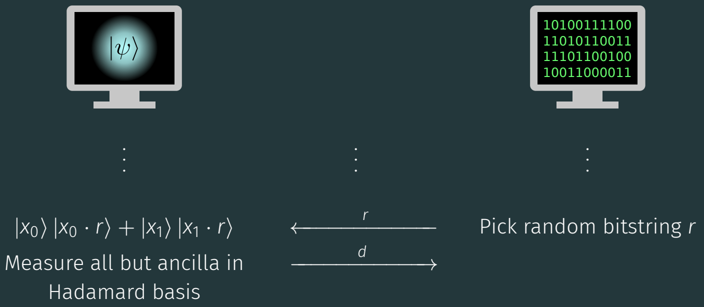
Measure all but ancilla in Hadamard basis $\xrightarrow{\qquad d \qquad}$

Single-qubit state: $|\uparrow\rangle$ or $|\downarrow\rangle$ if $x_0 \cdot r = x_1 \cdot r$, otherwise $|\leftarrow\rangle$ or $|\rightarrow\rangle$.

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."



$\vdots$ $\vdots$ $\vdots$

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

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

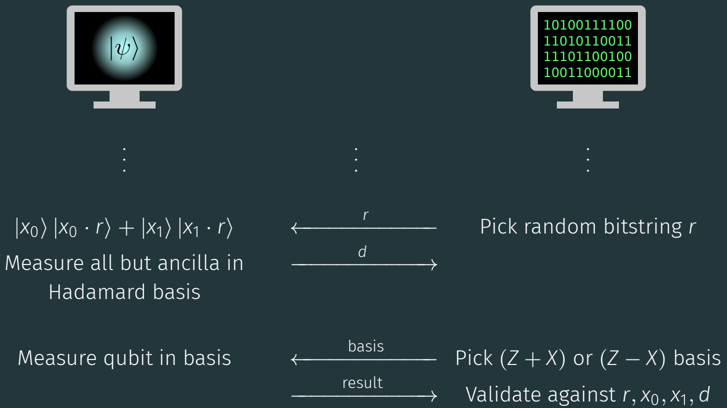
Single-qubit state: $|{\uparrow}\rangle$ or $|{\downarrow}\rangle$ if $x_0 \cdot r = x_1 \cdot r$, otherwise $|{\leftarrow}\rangle$ or $|{\rightarrow}\rangle$.
Polarization hidden via:

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.”



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

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

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

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:

Soundness (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:

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

Completeness (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:

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

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

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

Soundness (classical bound): $p_s + 4p_{\mathrm{CHSH}} - 4 < \mathrm{negl}(n)$

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

$p_s + 4p_{\mathrm{CHSH}} - 4 = \sqrt{2} - 1 \approx$ **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

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
    - Required for multi-round classical interaction

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
    - Required for multi-round classical interaction
    - U. Maryland experiment: first implementation in trapped ions!

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
    - Required for multi-round classical interaction
    - U. Maryland experiment: first implementation in trapped ions!
- Fidelity requirement

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction
  - U. Maryland experiment: first implementation in trapped ions!
- Fidelity requirement
  - High fidelity needed to pass classical bound

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction
  - U. Maryland experiment: first implementation in trapped ions!
- Fidelity requirement
  - High fidelity needed to pass classical bound
  - Postselection scheme allows passing with arbitrarily low fidelities

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction
  - U. Maryland experiment: first implementation in trapped ions!
- Fidelity requirement
  - High fidelity needed to pass classical bound
  - Postselection scheme allows passing with arbitrarily low fidelities
- Circuit sizes

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction
  - U. Maryland experiment: first implementation in trapped ions!
- Fidelity requirement
  - High fidelity needed to pass classical bound
  - Postselection scheme allows passing with arbitrarily low fidelities
- Circuit sizes
  - Need to implement public-key crypto. on a superposition

See arXiv:2104.00687 for details

Challenges for implementation

- Partial measurement
  - Required for multi-round classical interaction
  - U. Maryland experiment: first implementation in trapped ions!
- Fidelity requirement
  - High fidelity needed to pass classical bound
  - Postselection scheme allows passing with arbitrarily low fidelities
- Circuit sizes
  - Need to implement public-key crypto. on a superposition
  - Measurement scheme removes need for *reversibility* in quantum circuits—significant efficiency gains

See arXiv:2104.00687 for details

TCF constructions

TCF	A.H.C.B.	Gate count	n for hardness
LWE [1]	✓	$\mathcal{O}(n^2 \log^2 n)$	10^4
Ring-LWE [2]	✗	$\mathcal{O}(n \log^2 n)$	10^3
$x^2 \bmod N$ [3]	✗	$\mathcal{O}(n \log n)$	10^3
DDH [3]	✗	$\mathcal{O}(n^3 \log^2 n)$	10^2

A.H.C.B. = ”adaptive hard core bit”

[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)

TCF constructions

TCF	A.H.C.B.	Gate count	n for hardness
LWE [1]	✓	$\mathcal{O}(n^2 \log^2 n)$	10^4
Ring-LWE [2]	✗	$\mathcal{O}(n \log^2 n)$	10^3
$x^2 \bmod N$ [3]	✗	$\mathcal{O}(n \log n)$	10^3
DDH [3]	✗	$\mathcal{O}(n^3 \log^2 n)$	10^2

A.H.C.B. = ”adaptive hard core bit”

Remarks:

- Removing adaptive hardcore bit requirement helps!

[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)

TCF constructions

TCF	A.H.C.B.	Gate count	n for hardness
LWE [1]	✓	$\mathcal{O}(n^2 \log^2 n)$	10^4
Ring-LWE [2]	✗	$\mathcal{O}(n \log^2 n)$	10^3
$x^2 \bmod N$ [3]	✗	$\mathcal{O}(n \log n)$	10^3
DDH [3]	✗	$\mathcal{O}(n^3 \log^2 n)$	10^2

A.H.C.B. = "adaptive hard core bit"

Remarks:

- Removing adaptive hardcore bit requirement helps!
- Can't just plug in n—constant factors

[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)

$x^2 \bmod N$

$$y = x^2 \bmod N \text{ with } N = pq$$

Each y has 4 roots $(x_0, x_1, -x_0, -x_1)$.

$x^2 \bmod N$

$$y = x^2 \bmod N \text{ with } N = pq$$

Each y has 4 roots $(x_0, x_1, -x_0, -x_1)$. Set domain to $[0, N/2]$ to make it 2-to-1

$x^2 \bmod N$

$$y = x^2 \bmod N \text{ with } N = pq$$

Each y has 4 roots $(x_0, x_1, -x_0, -x_1)$. Set domain to $[0, N/2]$ to make it 2-to-1

- Finding a claw as hard as factoring N

$x^2 \bmod N$

$$y = x^2 \bmod N \text{ with } N = pq$$

Each y has 4 roots $(x_0, x_1, -x_0, -x_1)$. Set domain to $[0, N/2]$ to make it 2-to-1

- Finding a claw as hard as factoring N
- Features:
    - Simple to implement, asymptotically fast algorithms
    - Classical hardness in practice extremely well studied

$x^2 \bmod N$

$$y = x^2 \bmod N \text{ with } N = pq$$

Each y has 4 roots $(x_0, x_1, -x_0, -x_1)$. Set domain to $[0, N/2]$ to make it 2-to-1

- Finding a claw as hard as factoring N
- Features:
    - Simple to implement, asymptotically fast algorithms
    - Classical hardness in practice extremely well studied
- $\mathcal{O}(n \log n \log \log n)$ Schonhage-Strassen multiplication seems out of reach, but

$x^2 \bmod N$

$$y = x^2 \bmod N \text{ with } N = pq$$

Each y has 4 roots $(x_0, x_1, -x_0, -x_1)$. Set domain to $[0, N/2]$ to make it 2-to-1

- Finding a claw as hard as factoring N
- Features:
  - Simple to implement, asymptotically fast algorithms
  - Classical hardness in practice extremely well studied
- $\mathcal{O}(n \log n \log \log n)$ Schonhage-Strassen multiplication seems out of reach, but
- $\mathcal{O}(n^{1.58})$ Karatsuba mult. beats naive $\mathcal{O}(n^2)$ alg. at $n \sim 100$ (much earlier than in the classical case!)

$x^2 \bmod N$

$$y = x^2 \bmod N \text{ with } N = pq$$

Each y has 4 roots $(x_0, x_1, -x_0, -x_1)$. Set domain to $[0, N/2]$ to make it 2-to-1

- Finding a claw as hard as factoring N
- Features:
    - Simple to implement, asymptotically fast algorithms
    - Classical hardness in practice extremely well studied
- $\mathcal{O}(n \log n \log\log n)$ Schonhage-Strassen multiplication seems out of reach, but
- $\mathcal{O}(n^{1.58})$ Karatsuba mult. beats naive $\mathcal{O}(n^2)$ alg. at $n \sim 100$ (much earlier than in the classical case!)

Q. advantage in 10^6 Toffoli gates

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: linear algebra in the exponent

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: linear algebra in the exponent

Gen(1^λ)

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g

[1] Peikert, Waters. "Lossy trapdoor functions and their applications" (2008)

[2] Freeman, Goldreich, Klitz, Rosen, Segev. "More constructions of lossy and correlation-secure trapdoor functions" (2010)

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: **linear algebra in the exponent**

$\mathsf{Gen}(1^\lambda)$

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $\boldsymbol{M} \in \mathbb{Z}_q^{k \times k}$ for $k > \log q$

[1] Peikert, Waters. "Lossy trapdoor functions and their applications" (2008)

[2] Freeman, Goldreich, Klitz, Rosen, Segev. "More constructions of lossy and correlation-secure trapdoor functions" (2010)

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: linear algebra in the exponent

$\mathsf{Gen}(1^\lambda)$

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $M \in \mathbb{Z}_q^{k \times k}$ for $k > \log q$
3. Compute $g^M = (g^{M_{ij}}) \in \mathbb{G}^{k \times k}$

[1] Peikert, Waters. “Lossy trapdoor functions and their applications” (2008)

[2] Freeman, Goldreich, Klitz, Rosen, Segev. “More constructions of lossy and correlation-secure trapdoor functions” (2010)

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: **linear algebra in the exponent**

$\mathsf{Gen}(1^\lambda)$

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $\boldsymbol{M} \in \mathbb{Z}_q^{k \times k}$ for $k > \log q$
3. Compute $g^{\boldsymbol{M}} = (g^{\boldsymbol{M}_{ij}}) \in \mathbb{G}^{k \times k}$
4. Return $pk = (g^M)$, $sk = (g, M)$

[1] Peikert, Waters. "Lossy trapdoor functions and their applications" (2008)

[2] Freeman, Goldreich, Klitz, Rosen, Segev. "More constructions of lossy and correlation-secure trapdoor functions" (2010)

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: **linear algebra in the exponent**

$pk = (g^{M})$, $sk = (g, M)$. On input $\mathbf{x} \in \{0, 1\}^{k}$:

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: linear algebra in the exponent

$pk = (g^M)$, $sk = (g, M)$. On input $\mathbf{x} \in \{0,1\}^k$:

Evaluation: $f(x) = g^{\mathbf{Mx}}$

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: linear algebra in the exponent

$pk = (g^{M})$, $sk = (g, M)$. On input $\mathbf{x} \in \{0,1\}^k$:

Evaluation: $f(x) = g^{\mathbf{Mx}}$

Inversion: $f^{-1}(f(x), M) = g^{\mathbf{M}^{-1}\mathbf{Mx}} = g^{\mathbf{x}}$

Easy to find $\mathbf{x}$ from $g^{\mathbf{x}}$ by brute force

---

Trapdoor from Decisional Diffie-Hellman (DDH)

Trapdoor functions from DDH [1, 2]: **linear algebra in the exponent**

$pk = (g^M)$, $sk = (g, M)$. On input $\mathbf{x} \in \{0,1\}^k$:

Evaluation: $f(x) = g^{\mathbf{Mx}}$

Inversion: $f^{-1}(f(x), M) = g^{\mathbf{M}^{-1}\mathbf{Mx}} = g^{\mathbf{x}}$

Easy to find $\mathbf{x}$ from $g^{\mathbf{x}}$ by brute force

---

Security proof: Given $g^{\mathbf{M}}$, DDH hides rank of $\mathbf{M}$. Inversion would imply algorithm to determine if $\mathbf{M}$ is full rank.

[1] Peikert, Waters. "Lossy trapdoor functions and their applications" (2008)

[2] Freeman, Goldreich, Klitz, Rosen, Segev. "More constructions of lossy and correlation-secure trapdoor functions" (2010)

TCF from DDH

$\mathsf{Gen}(1^\lambda)$

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g

TCF from DDH

Gen(1^λ)

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $M \in \mathbb{Z}_q^{k \times k}$ for $k > \log q$

TCF from DDH

$\mathsf{Gen}(1^\lambda)$

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $M \in \mathbb{Z}_q^{k\times k}$ for $k > \log q$
3. Compute $g^M = (g^{M_{ij}}) \in \mathbb{G}^{k\times k}$

TCF from DDH

$\mathsf{Gen}(1^\lambda)$

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $\boldsymbol{M} \in \mathbb{Z}_q^{k \times k}$ for $k > \log q$
3. Compute $g^{\boldsymbol{M}} = (g^{\boldsymbol{M}_{ij}}) \in \mathbb{G}^{k \times k}$
4. Choose $\boldsymbol{s} \in \{0, 1\}^k$

TCF from DDH

Gen(1^λ)

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $\boldsymbol{M} \in \mathbb{Z}_q^{k \times k}$ for $k > \log q$
3. Compute $g^{\boldsymbol{M}} = (g^{\boldsymbol{M}_{ij}}) \in \mathbb{G}^{k \times k}$
4. Choose $\boldsymbol{s} \in \{0, 1\}^k$
5. Return $pk = (g^M, g^{Ms})$, $sk = (g, M, s)$

TCF from DDH

Gen(1^λ)

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $\boldsymbol{M} \in \mathbb{Z}_q^{k \times k}$ for $k > \log q$
3. Compute $g^{\boldsymbol{M}} = (g^{\boldsymbol{M}_{ij}}) \in \mathbb{G}^{k \times k}$
4. Choose $\boldsymbol{s} \in \{0, 1\}^k$
5. Return $pk = (g^M, g^{Ms})$, $sk = (g, M, s)$

TCF from DDH

$\mathrm{Gen}(1^\lambda)$

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $\boldsymbol{M} \in \mathbb{Z}_q^{k\times k}$ for $k > \log q$
3. Compute $g^{\boldsymbol{M}} = (g^{\boldsymbol{M}_{ij}}) \in \mathbb{G}^{k\times k}$
4. Choose $\boldsymbol{s} \in \{0,1\}^k$
5. Return $pk = (g^M, g^{Ms})$, $sk = (g, M, s)$

Evaluation:

Let $d \sim \mathcal{O}(k^2)$. Define two functions $f_b : \mathbb{Z}_d^k \to \mathbb{G}^k$:

$$f_0(x) = g^{Mx} \qquad\qquad f_1(x) = g^{Mx}g^{Ms} = g^{M(x+s)}$$

TCF from DDH

Gen(1^λ)

1. Choose group $\mathbb{G}$ of order $q \sim \mathcal{O}(2^\lambda)$, and generator g
2. Choose random invertible $\boldsymbol{M} \in \mathbb{Z}_q^{k\times k}$ for $k > \log q$
3. Compute $g^{\boldsymbol{M}} = (g^{\boldsymbol{M}_{ij}}) \in \mathbb{G}^{k\times k}$
4. Choose $\boldsymbol{s} \in \{0,1\}^k$
5. Return $pk = (g^M, g^{Ms})$, $sk = (g, M, s)$

Evaluation:

Let $d \sim \mathcal{O}(k^2)$. Define two functions $f_b : \mathbb{Z}_d^k \to \mathbb{G}^k$:

$$f_0(x) = g^{Mx} \qquad\qquad f_1(x) = g^{Mx}g^{Ms} = g^{M(x+s)}$$

Inversion: $f^{-1}(f_0(x), M) = g^{\boldsymbol{M}^{-1}\boldsymbol{M}\boldsymbol{x}} = g^{\boldsymbol{x}}$ (poly-time brute force)

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

TCF from DDH: does it help?

- Via elliptic curves, can significantly reduce space requirement

TCF from DDH: does it help?

- Via elliptic curves, can significantly reduce space requirement
- But quantum circuit for group operation is **complicated**

TCF from DDH: does it help?

- Via elliptic curves, can significantly reduce space requirement
- But quantum circuit for group operation is **complicated**
- Need to perform as many group operations as Shor's algorithm!

TCF from DDH: does it help?

- Via elliptic curves, can significantly reduce space requirement
- But quantum circuit for group operation is **complicated**
- Need to perform as many group operations as Shor's algorithm!
- Reversible Euclidean algorithm is hard, maybe irreversible optimization can help?

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?)
- **Remove need for trapdoor** (hash functions?)

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?)
- **Remove need for trapdoor** (hash functions?)
- Sub-exponential verification?

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?)
- **Remove need for trapdoor** (hash functions?)
- Sub-exponential verification?

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

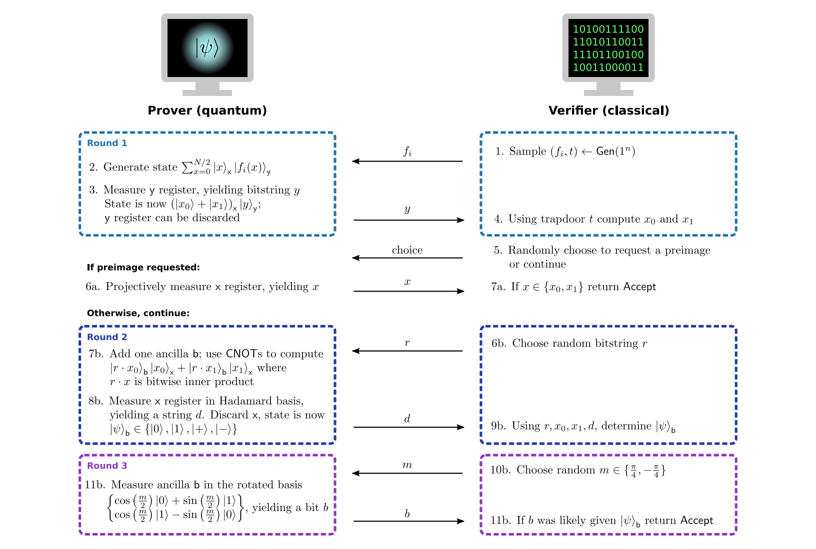
"Box-adjacent" ideas:

- Explore other protocols (fix IQP and make it fast?)
- **Remove need for trapdoor** (hash functions?)
- Sub-exponential verification?

Way outside the box?

Backup!

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

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

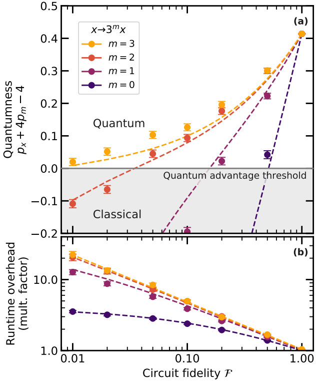
How to deal with high fidelity requirement? Need $\sim 83\%$ fidelity in general to pass.

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 $x^2 \bmod N \Rightarrow (kx)^2 \bmod k^2 N$

Partial measurements in the lab

TIQI

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Prof. Christopher Monroe

Dr. Daiwei Zhu

Dr. Crystal Noel

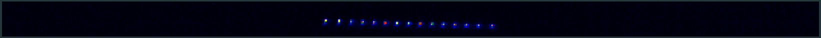
and others!

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

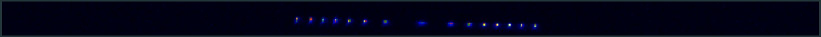


Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:



Partial measurements in the lab

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

TIQI

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

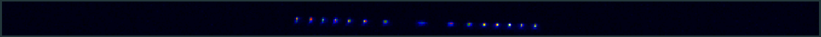
Partial measurement:

Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:

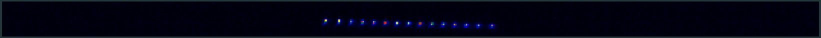


Partial measurements in the lab

Trapped Ion Quantum Information lab at U. Maryland

Working on demonstration of protocols in trapped ions!

Partial measurement:



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!

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

Improving circuit sizes

Most demanding step in all these protocols: evaluating TCF

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

Getting rid of 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!

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

but they are recursive and hard to make reversible.

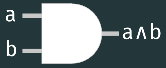
Protocol allows us to make circuits irreversible!

Technique: taking out the garbage

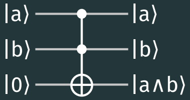
Goal: $\mathcal{U}_f \left|x\right\rangle \left|0^{\otimes n}\right\rangle = \left|x\right\rangle \left|f(x)\right\rangle$

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity



Classical AND



Quantum AND (Toffoli)

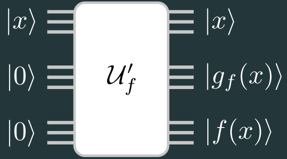
Technique: taking out the garbage

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 $g_f(x)$:



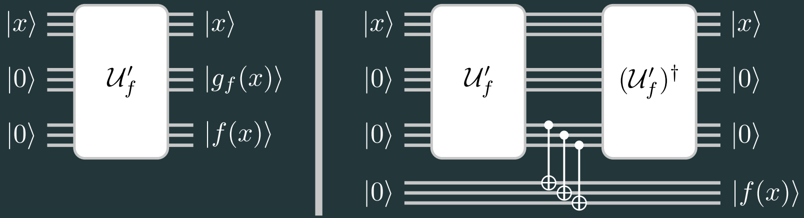
Technique: taking out the garbage

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 $g_f(x)$:



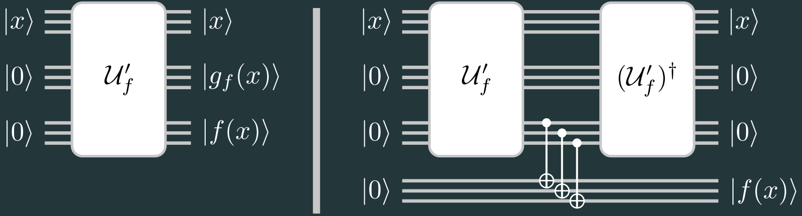
Technique: taking out the garbage

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

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!

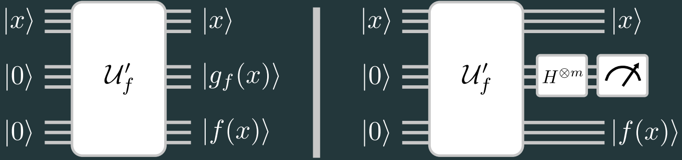
Technique: taking out the garbage

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

When converting classical circuits to quantum:

Garbage bits: extra entangled outputs due to unitarity

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



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 $x^2 \bmod N$ costs only 10^6 Toffoli gates.