Joint Center for Quantum Information and Computer Science

University of Maryland 1856

HIDING SECRETS IN IQP CIRCUITS

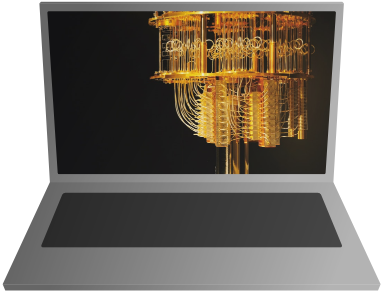
A drama in three acts

Dominik Hangleiter

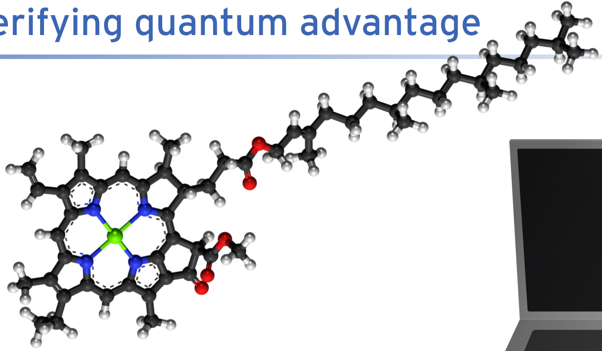
with David Gross

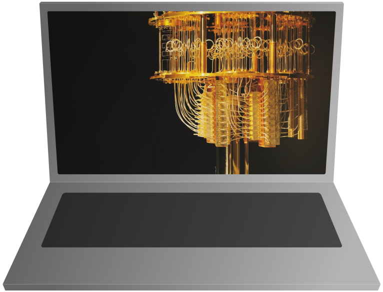
Arlington, June 11, 2024

Verifying quantum advantage

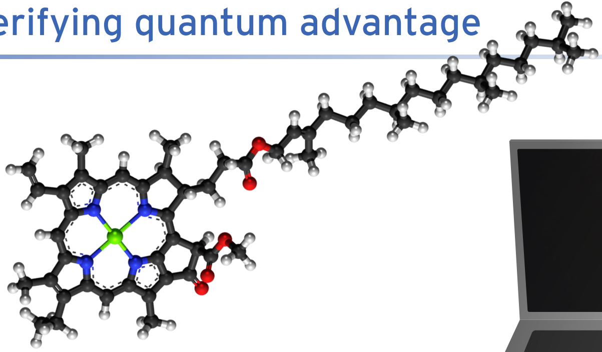


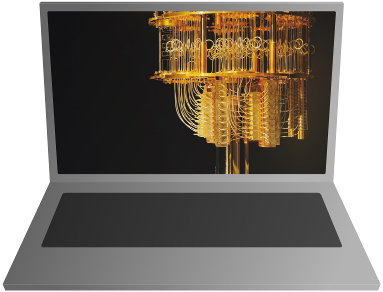
Verifying quantum advantage





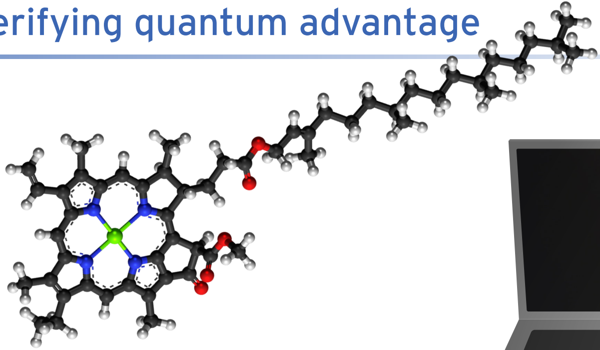
Verifying quantum advantage

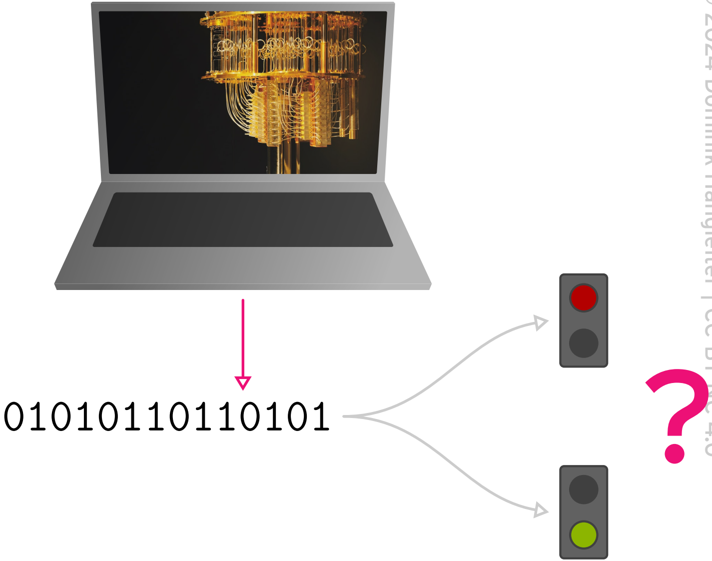




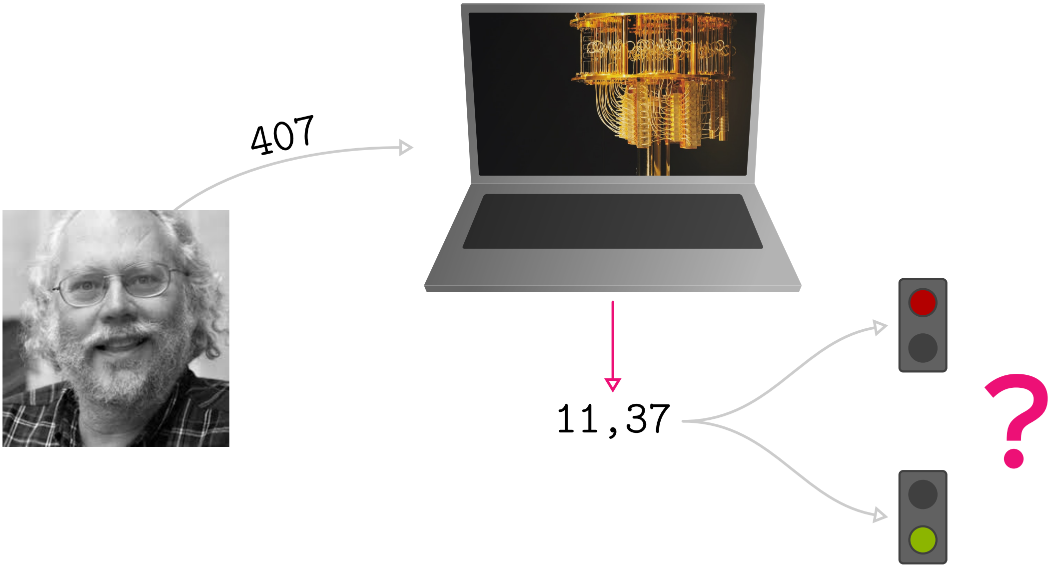
01010110110101

Verifying quantum advantage





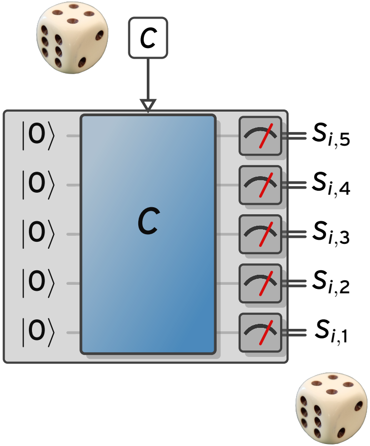
Verifying quantum advantage



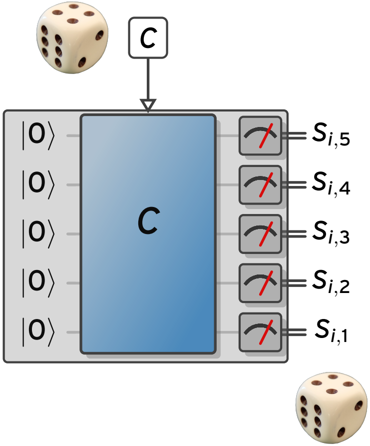
Quantum random sampling

 $\boxed{C} \in \{C_0, \ldots, C_N\}$

Quantum random sampling



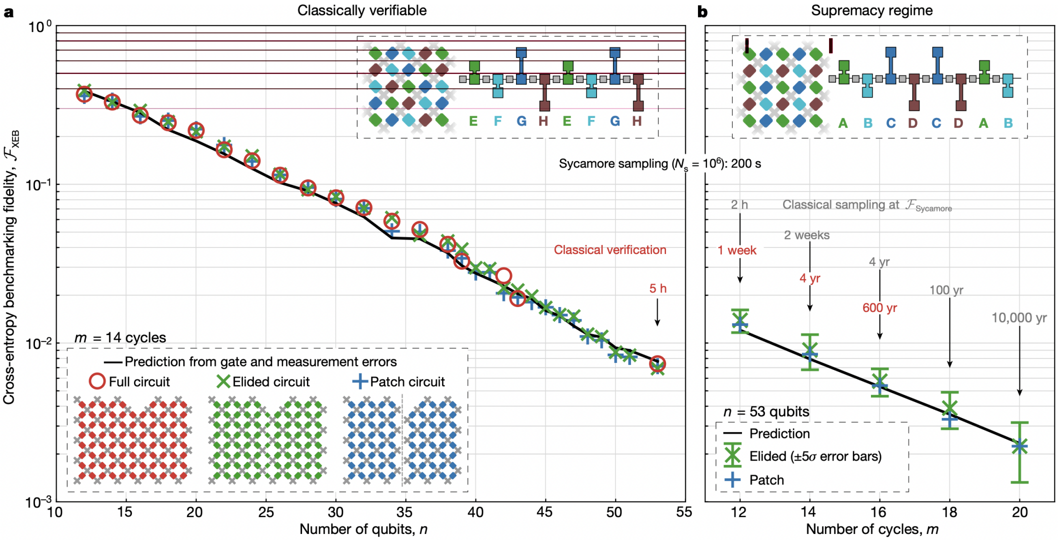
Quantum random sampling



Classical simulations are *provably* inefficient.

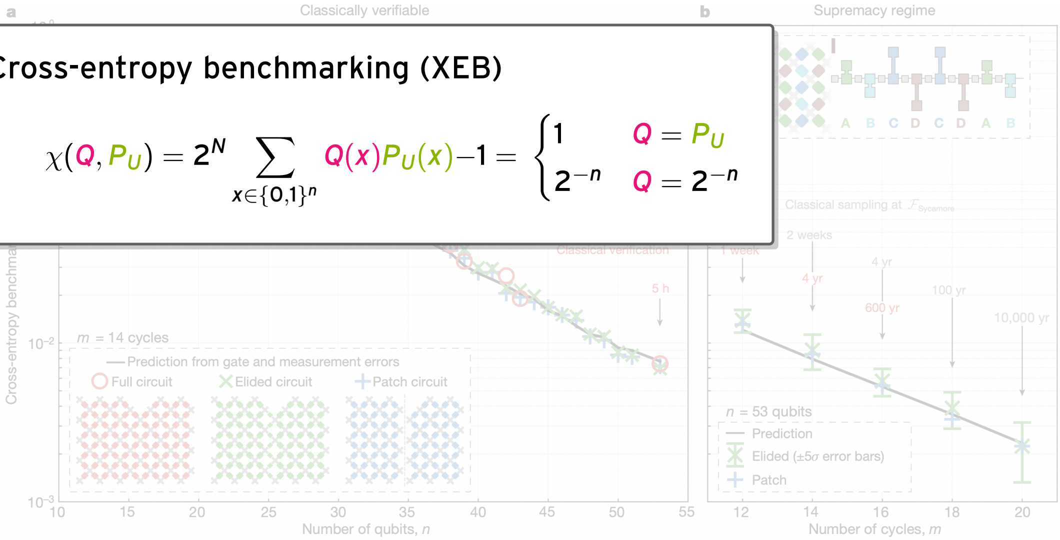
Review: DH, Eisert, RMP (2023)

Verifying quantum sampling



Arute *et al*, Nature (2019)

Verifying quantum sampling

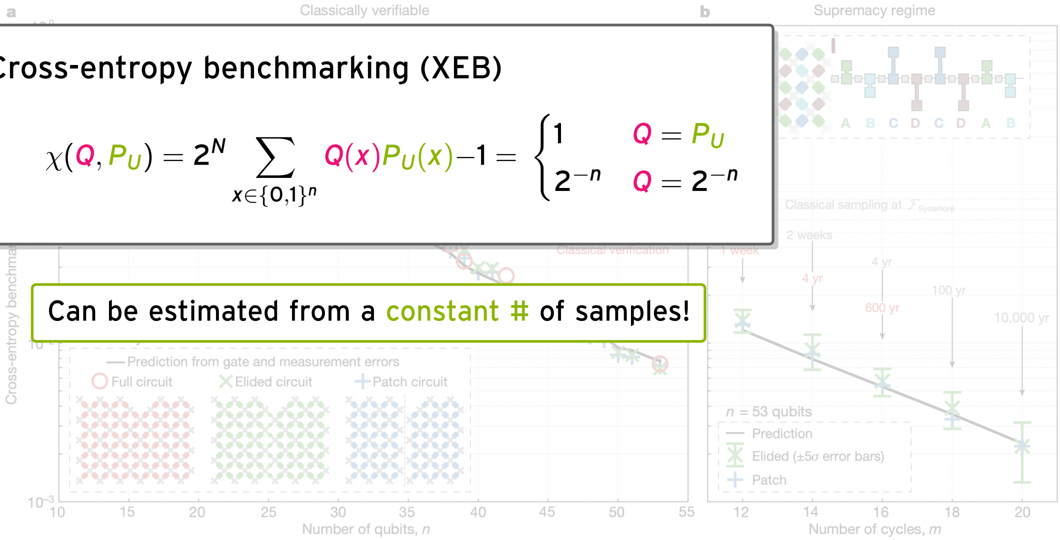


Cross-entropy benchmarking (XEB)

$$\chi(Q, P_U) = 2^N \sum_{x \in \{0,1\}^n} Q(x) P_U(x) - 1 = \begin{cases} 1 & Q = P_U \\ 2^{-n} & Q = 2^{-n} \end{cases}$$

Arute *et al*, Nature (2019)

Verifying quantum sampling



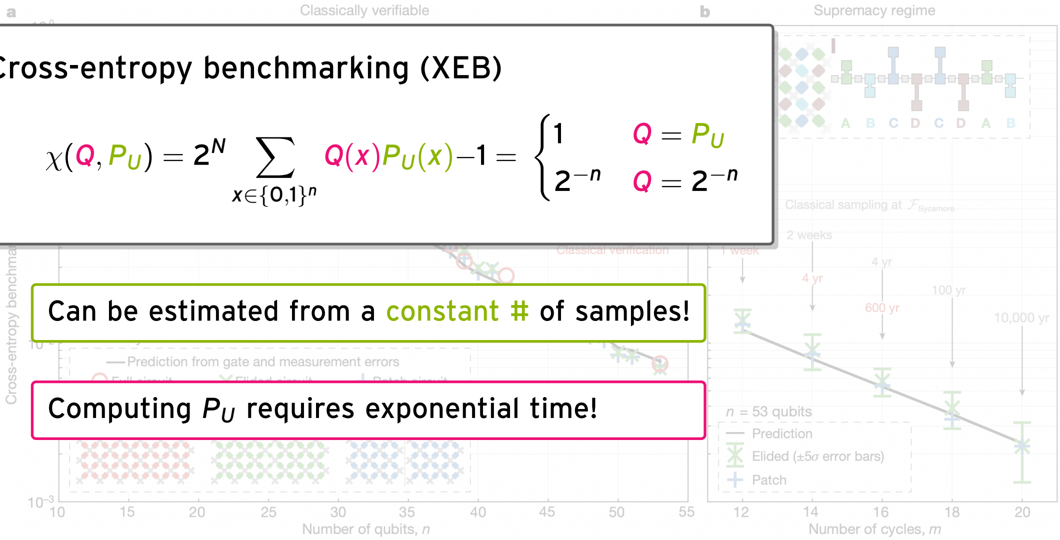
Cross-entropy benchmarking (XEB)

$$\chi(Q, P_U) = 2^N \sum_{x \in \{0,1\}^n} Q(x) P_U(x) - 1 = \begin{cases} 1 & Q = P_U \\ 2^{-n} & Q = 2^{-n} \end{cases}$$

Can be estimated from a constant # of samples!

Arute *et al*, Nature (2019)

Verifying quantum sampling



Cross-entropy benchmarking (XEB)

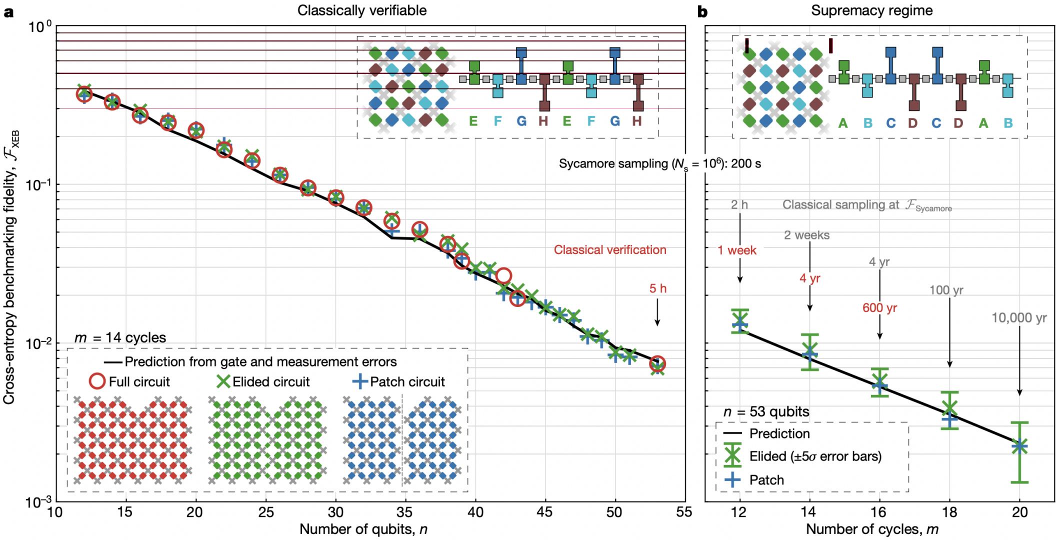
$$\chi(Q, P_U) = 2^N \sum_{x \in \{0,1\}^n} Q(x) P_U(x) - 1 = \begin{cases} 1 & Q = P_U \\ 2^{-n} & Q = 2^{-n} \end{cases}$$

Can be estimated from a constant # of samples!

Computing P_U requires exponential time!

Arute *et al*, Nature (2019)

Verifying quantum sampling



Arute *et al*, Nature (2019)

Can we efficiently verify quantum sampling?

ACT I

Dan and Mick have an idea

X-programs

X program [SB09]

- Angle θ
- $\mathbf{P} \in \{0,1\}^{m \times n}$
- $H_{\mathbf{P}} = \sum_i \left(\prod_j X_j^{\mathbf{P}_{i,j}} \right)$

X-programs

X program [SB09]

- Angle θ
- $\mathbf{P} \in \{0,1\}^{m \times n}$
- $H_{\mathbf{P}} = \sum_i \left(\prod_j X_j^{\mathbf{P}_{i,j}} \right)$

$$C = e^{i\theta H_{\mathbf{P}}}$$

X-programs

X program [SB09]

- Angle θ
- $\mathbf{P} \in \{0,1\}^{m \times n}$
- $H_{\mathbf{P}} = \sum_i \left(\prod_j X_j^{\mathbf{P}_{i,j}} \right)$

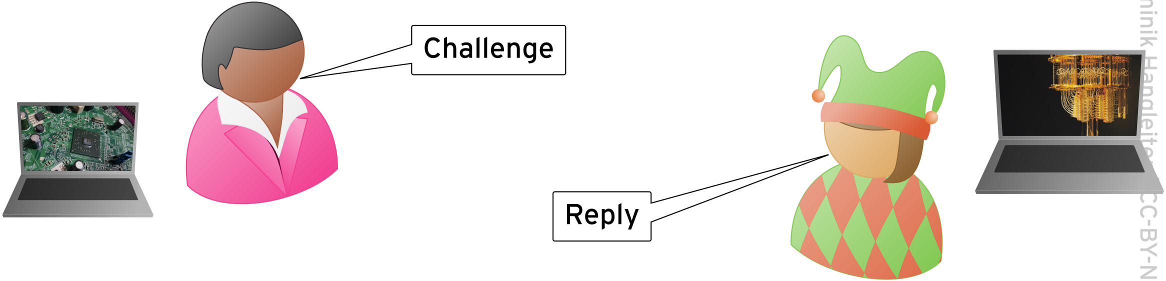
$$C = e^{i\theta H_{\mathbf{P}}}$$

Example

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \qquad H_{\mathbf{P}} = X_2X_3 + X_1X_2X_3 + X_1X_2X_3X_4 + X_2 + X_1X_3X_4$$

[SB09] Shepherd & Bremner (2009) [She10] Shepherd, arXiv:1005.1744 [BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

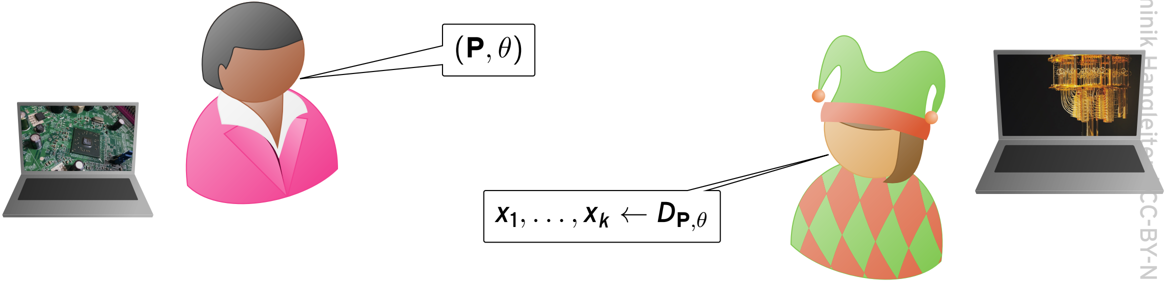
Hiding a secret in the output of an X-program



Hiding a secret in the output of an X-program

→ Design an X program $(\mathbf{P}, \theta)$ with a secret s.

Prepare $e^{i\theta H_{\mathbf{P}}}|0\rangle$ and measure. ←

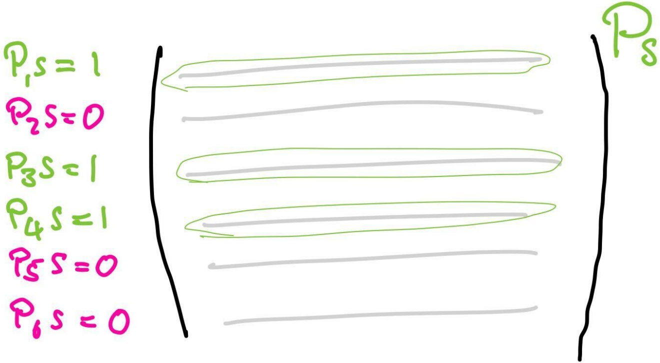


→ Compute $\frac{1}{k}\sum_k x_k \cdot s \approx \Pr[x \cdot s = 0]$.

Dan and Mick's tricks

The double angle trick [SB09,She10]
Fourier coefficients are given by the zero-amplitude of a different *X*-program with double angle

$$\beta_s = \langle Z_s \rangle = \langle 0 | e^{i 2\theta H_{\mathbf{P}_s}} | 0 \rangle, \qquad \text{where } (\mathbf{P}_s)_i = \mathbf{P}_i \text{ iff } \mathbf{P}_i \cdot s = 1.$$



Dan and Mick's tricks

The double angle trick [SB09,She10]
Fourier coefficients are given by the zero-amplitude of a different X-program with double angle

$$\beta_s = \langle Z_s \rangle = \langle 0 | e^{i2\theta H_{\mathbf{P}_s}} | 0 \rangle, \qquad \text{where } (\mathbf{P}_s)_i = \mathbf{P}_i \text{ iff } \mathbf{P}_i \cdot s = 1.$$

- For $\theta = \pi/4$, an X-program is a Clifford circuit.
- Can compute Fourier coefficients for hard circuits with $\theta = \pi/8$.
- Sampling from random X programs with $\theta = \pi/8$ is classically hard.

Dan and Mick's tricks

The double angle trick [SB09,She10]
Fourier coefficients are given by the zero-amplitude of a different X-program with double angle

$$\beta_s = \langle Z_s \rangle = \langle 0 | e^{i2\theta H_{\mathbf{P}_s}} | 0 \rangle, \qquad \text{where } (\mathbf{P}_s)_i = \mathbf{P}_i \text{ iff } \mathbf{P}_i \cdot s = 1.$$

- For $\theta = \pi/4$, an X-program is a Clifford circuit.
- Can compute Fourier coefficients for hard circuits with $\theta = \pi/8$.
- Sampling from random X programs with $\theta = \pi/8$ is classically hard.

The coding theory trick [SB09,She10]

$$\langle 0 | e^{i\pi/4 H_{\mathbf{P}}} | 0 \rangle = \begin{cases} 2^{-\operatorname{rank}(\mathbf{P}^T\mathbf{P})/2} & \operatorname{col}(\mathbf{P}) \cap \operatorname{col}(\mathbf{P})^\perp \text{ is doubly even} \\ 0 & \text{else} \end{cases}$$

Dan and Mick's tricks

The double angle trick [SB09,She10]

Fourier coefficients are given by the zero-amplitude of a different X-program with double angle

$\beta_s = \langle Z_s \rangle = \langle 0 | e^{i 2\theta H_{\mathbf{P}_s}} | 0 \rangle$ where $(\mathbf{P}_s)_i = \mathbf{P}_i$ iff $\mathbf{P}_i \cdot s = 1$.

For random $\mathbf{P}$, $\mathrm{rank}(\mathbf{P}^T\mathbf{P}) \sim n$

→ For most $\mathbf{s}$, $\beta_{\mathbf{s}} \lesssim 2^{-n}$

…Clifford circuit.

…s for hard circuits with $\theta = \pi/8$.

Sampling from random X programs with $\theta = \pi/8$ is classically hard.

The coding theory trick [SB09,She10]

$$\langle 0 | e^{i\pi/4 H_{\mathbf{P}}} | 0 \rangle = \begin{cases} 2^{-\mathrm{rank}(\mathbf{P}^T\mathbf{P})/2} & \mathrm{col}(\mathbf{P}) \cap \mathrm{col}(\mathbf{P})^\perp \text{ is doubly even} \\ 0 & \text{else} \end{cases}$$

Dan and Mick's tricks

The double angle trick [SB09,She10]
Fourier coefficients are given by the zero-a… program with double angle

$$\beta_s = \langle Z_s \rangle = \langle 0|e^{i2\theta H_{\mathbf{P}_s}}|0\rangle$$ wh…

For random $\mathbf{P}$, $\mathrm{rank}(\mathbf{P}^T\mathbf{P}) \sim n$

→ For most $\mathbf{s}$, $\beta_{\mathbf{s}} \lesssim 2^{-n}$

Goal
Design $\mathbf{P}$ such that $\mathbf{P}_s^T\mathbf{P}_s$ has large kernel for a secret $\mathbf{s}$.

…lifford circuit.

…s for hard circuits with $\theta = \pi/8$.

Sampling from random X programs with $\theta = \pi/8$ is classically hard.

The coding theory trick [SB09,She10]

$$\langle 0|e^{i\pi/4 H_{\mathbf{P}}}|0\rangle = \begin{cases} 2^{-\mathrm{rank}(\mathbf{P}^T\mathbf{P})/2} & \mathrm{col}(\mathbf{P}) \cap \mathrm{col}(\mathbf{P})^{\perp} \text{ is doubly even} \\ 0 & \text{else} \end{cases}$$

Interlude: Geometry of the problem

→ Understand $\operatorname{rank}(\mathbf{P}_s^T \mathbf{P}_s)$

Interlude: Geometry of the problem

→ Understand $\text{rank}(\mathbf{P}_s^T \mathbf{P}_s)$

→ $\mathbf{P}_s^T \mathbf{P}_s$ is the Gram matrix describing the geometry of $\text{col}(\mathbf{P}_s)$

Interlude: Geometry of the problem

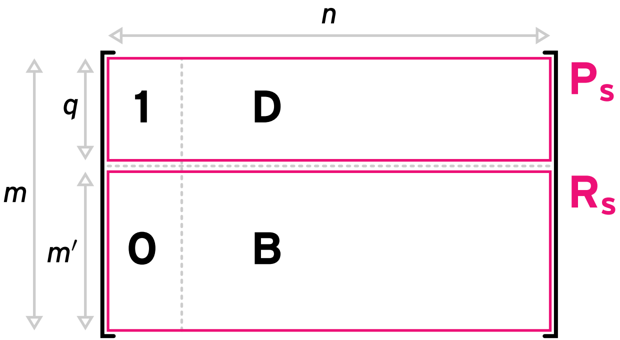
→ Understand $\operatorname{rank}(\mathbf{P}_\mathbf{s}^T \mathbf{P}_\mathbf{s})$

→ $\mathbf{P}_\mathbf{s}^T \mathbf{P}_\mathbf{s}$ is the Gram matrix describing the geometry of $\operatorname{col}(\mathbf{P}_\mathbf{s})$

→ $\mathbf{d} \in \ker \mathbf{P}_\mathbf{s}^T \mathbf{P}_\mathbf{s} \quad \Leftrightarrow \quad \mathbf{d} \in \operatorname{rad} \operatorname{col}(\mathbf{P}_\mathbf{s})$,

Radical of vector space V: $\operatorname{rad}(V) = V \cap V^\perp$

Hiding a secret in the output of an X-program



Hiding a secret in the output of an X-program

$$\mathbf{P} = \left[\begin{array}{c:c} \mathbf{1} & \mathbf{D} \\ \hdashline \mathbf{0} & \mathbf{B} \end{array}\right] \cdot \mathbf{X}$$

$$\boldsymbol{s} = \left[\begin{array}{c:c} 1, & 0, 0, \ldots \end{array}\right]$$

→ [BS09] choose **D** as a quadratic residue code (radical is doubly even).

Hiding a secret in the output of an X-program

$$\mathbf{P} = \left[\begin{array}{c:c} \mathbf{1} & \mathbf{D} \\ \hdashline \mathbf{0} & \mathbf{B} \end{array}\right] \cdot \mathbf{X}$$

$$\boldsymbol{s} = [\,1, \;\vdots\; 0, 0, \ldots\,]$$

→ [BS09] choose **D** as a quadratic residue code (radical is doubly even).

→ The output distribution of $(\mathbf{P}, \pi/8)$ has $\beta_{\mathbf{s}} = 1/\sqrt{2}$

Hiding a secret in the output of an X-program

$$\mathbf{P} = \Pi \cdot \begin{bmatrix} \mathbf{1} & \mathbf{D} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \cdot \mathbf{X}$$

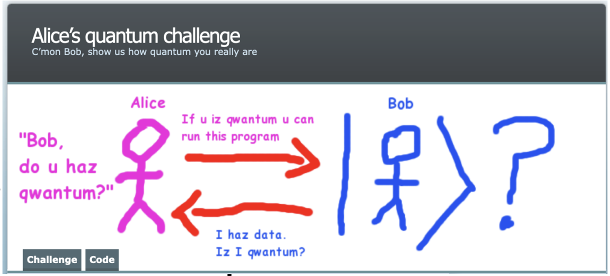
$$\boldsymbol{s} = \begin{bmatrix} 1, & 0, 0, \ldots \end{bmatrix} \cdot \mathbf{X}$$

Hiding a secret in the output of an X-program

$$\mathbf{P} = \Pi \cdot \left[\begin{array}{c|c} \mathbf{1} & \mathbf{D} \\ \hline \mathbf{0} & \mathbf{B} \end{array}\right] \cdot \mathbf{X}$$

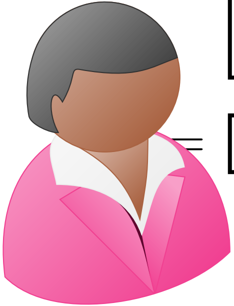
$$\mathbf{s} = \left[\begin{array}{c|c} 1, & 0, 0, \ldots \end{array}\right] \cdot \mathbf{X}$$

Hiding a secret in the output of an X-program



$$\mathbf{P} = \Pi \cdot \begin{bmatrix} \mathbf{1} & \mathbf{D} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$$

$$= [1,\ 0, 0, \ldots] \cdot \mathbf{X}$$

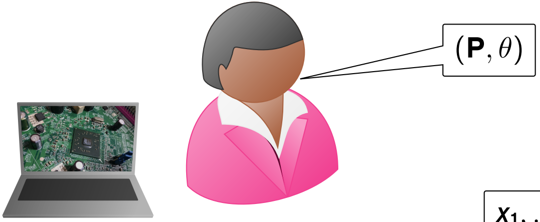


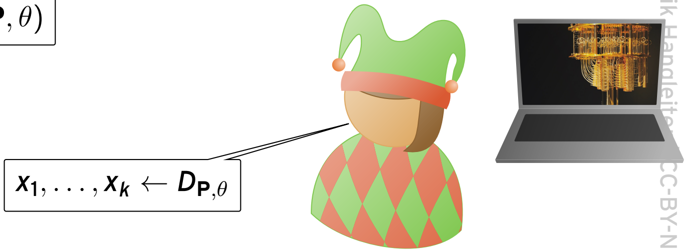
$$\beta_{\mathbf{s}} = 2 \Pr[x \cdot \mathbf{s} = 0] - 1$$

Hiding a secret in the output of an X-program

→ Design an X program $(\mathbf{P}, \theta)$ with a secret s.

Prepare $e^{i\theta H_{\mathbf{P}}}|0\rangle$ and measure. ←





→ Compute $\frac{1}{k}\sum_k x_k \cdot s \approx \Pr[x \cdot s = 0]$.

ACT II

Greg is a killjoy but IQP comes back

The Kahanamoku-Meyer attack

Greg's trick [Kah19]

$$\text{For } d \in \mathbb{F}_2^n : \mathbf{P}_s d \in \operatorname{rad}(\operatorname{col}(\mathbf{P}_s)) \quad \Rightarrow \quad s \in \ker(\mathbf{P}_d^T \mathbf{P}_d).$$

[Kah19] Kahanamoku-Meyer arXiv:1912.05547 [GH23] Gross & DH, arXiv:2312.10156

The Kahanamoku-Meyer attack

Greg's trick [Kah19]

$$\text{For } d \in \mathbb{F}_2^n : \mathbf{P}_s d \in \operatorname{rad}(\operatorname{col}(\mathbf{P_s})) \quad \Rightarrow \quad s \in \ker(\mathbf{P}_d^T \mathbf{P}_d).$$

Attack

1 Draw d randomly.

2 Iterate through the elements $t \in \ker \mathbf{G}_d$ and check if $\mathbf{P}_t$ generates a QRC.

[Kah19] Kahanamoku-Meyer arXiv:1912.05547 [GH23] Gross & DH, arXiv:2312.10156

The Kahanamoku-Meyer attack

Greg's trick [Kah19]

$$\text{For } d \in \mathbb{F}_2^n : \mathbf{P}_s d \in \mathrm{rad}(\mathrm{col}(\mathbf{P_s})) \quad \Rightarrow \quad s \in \ker(\mathbf{P}_d^T \mathbf{P}_d).$$

Attack

1 Draw d randomly.

2 Iterate through the elements $t \in \ker \mathbf{G}_d$ and check if $\mathbf{P}_t$ generates a QRC.

- With probability $2^{-\mathrm{rank}(\mathbf{P}_s^T \mathbf{P_s})}$ s lies in $\ker \mathbf{P_d}^T \mathbf{P_d}$.
- For the [SB09] QRC construction $\ker \mathbf{P_d}^T \mathbf{P_d}$ is typically small ($2^{n-m/2}$ elements).

[Kah19] Kahanamoku-Meyer arXiv:1912.05547 [GH23] Gross & DH, arXiv:2312.10156

The Kahanamoku-Meyer attack

Greg's trick [Kah19]

For $d \in \mathbb{F}_2^n$: $\mathbf{P}_s d \in \text{rad}(\text{col}(\mathbf{P_s}))$ $\Rightarrow$ $s \in \ker(\mathbf{P}_d^T\mathbf{P}_{\ldots}$

Attack

1 Draw d randomly.

2 Iterate through the el… $\mathbf{G}_d$ and check if $\mathbf{P}_t$ generates a QRC.

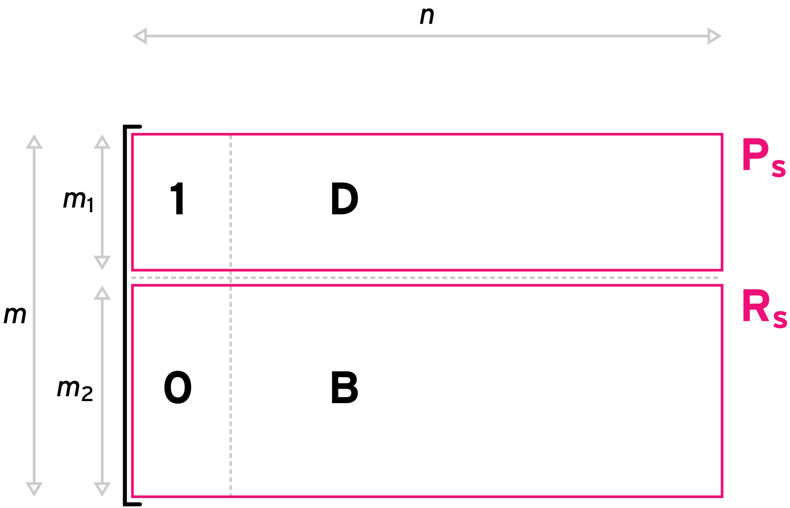
The SECRET is Bi1bHzjYxr0HYH401EJFBoXZbps4a54kH8f1rRgo/g==

…ility $2^{-\text{rank}(\mathbf{P_s}^T\mathbf{P_s})}$ s lies in $\ker \mathbf{P}_\mathbf{d}^T\mathbf{P}_\mathbf{d}$.

… [SB09] QRC construction $\ker \mathbf{P}_\mathbf{d}^T\mathbf{P}_\mathbf{d}$ is typically small ($2^{n-m/2}$ elements).

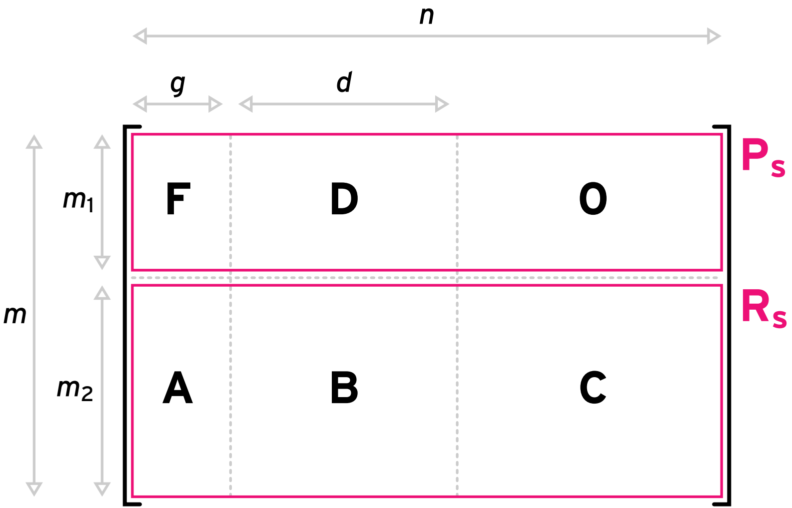
[Kah19] Kahanamoku-Meyer arXiv:1912.05547 [GH23] Gross & DH, arXiv:2312.10156

Removing structure from the secret hiding [BCJ23]



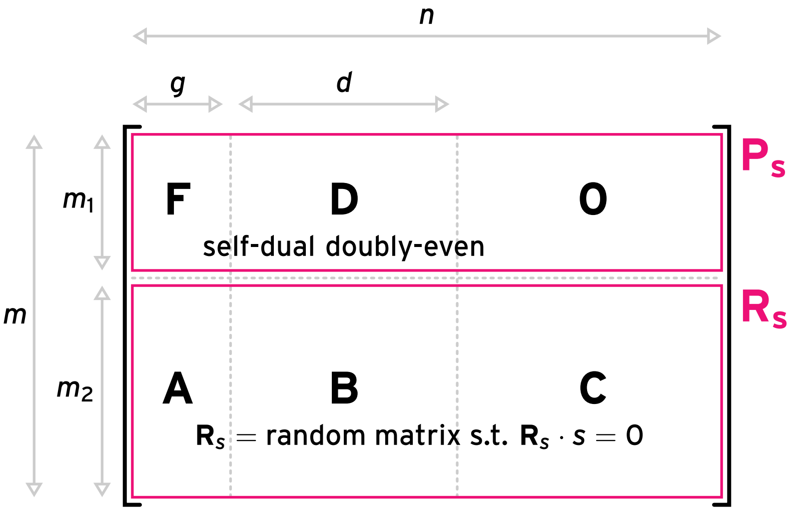
[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

Removing structure from the secret hiding [BCJ23]



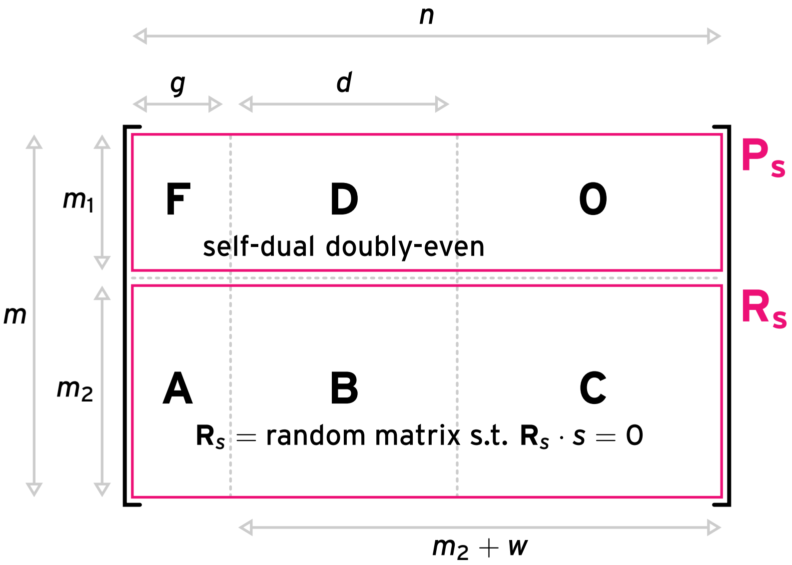
[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

Removing structure from the secret hiding [BCJ23]



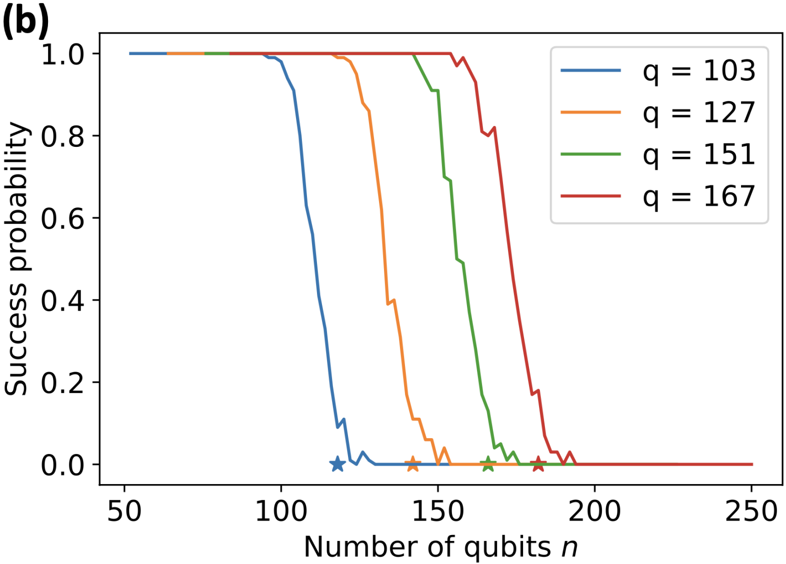
[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

Removing structure from the secret hiding [BCJ23]



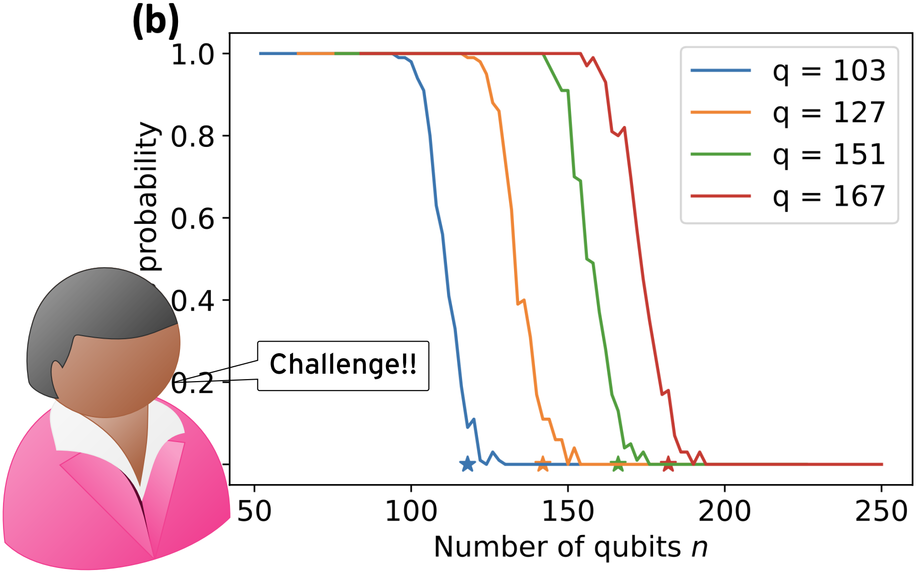
[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

Removing structure from the secret hiding



[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

Removing structure from the secret hiding



[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

ACT III

Hope for IQP is waning

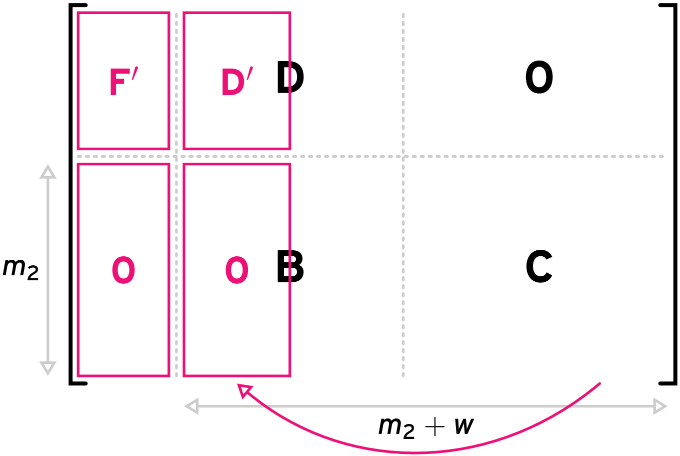
The radical attack

$$\begin{bmatrix} \mathbf{F} & \mathbf{D} & \mathbf{0} \\ \mathbf{A} & \mathbf{B} & \mathbf{C} \end{bmatrix}$$

- Range is unchanged under column operations.
- If range$[\mathbf{B}|\mathbf{C}] = \mathbb{F}_2^{m_2}$, can 'clear' columns below $[\mathbf{F}|\mathbf{D}]$.
- Elements of $\mathbf{D}'$ are in the radical of $\mathbf{H}$!
- The support of $\mathrm{col}(\mathbf{D}')$ determines the secret.

[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

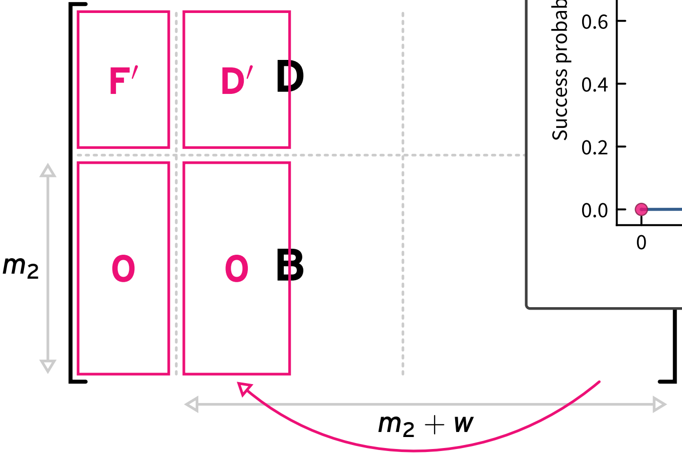
The radical attack

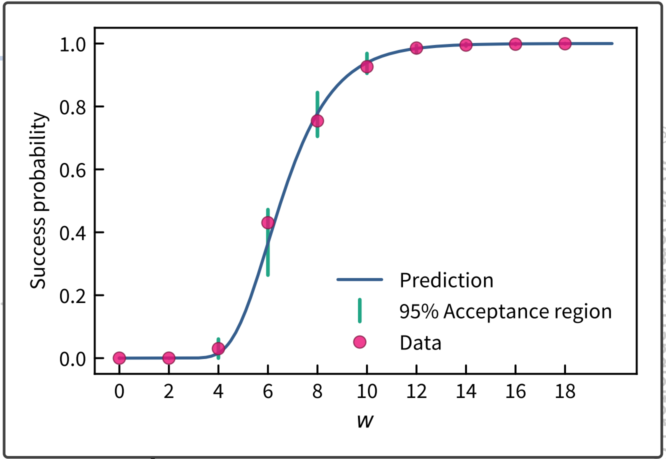


- Range is unchanged under column operations.
- If range$[\mathbf{B}|\mathbf{C}] = \mathbb{F}_2^{m_2}$, can 'clear' columns below $[\mathbf{F}|\mathbf{D}]$.
- Elements of $\mathbf{D}'$ are in the radical of **H**!
- The support of $\mathrm{col}(\mathbf{D}')$ determines the secret.

[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

The radical attack

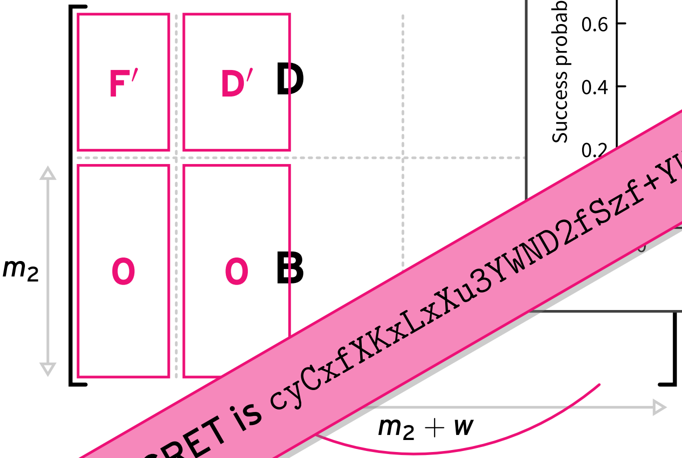


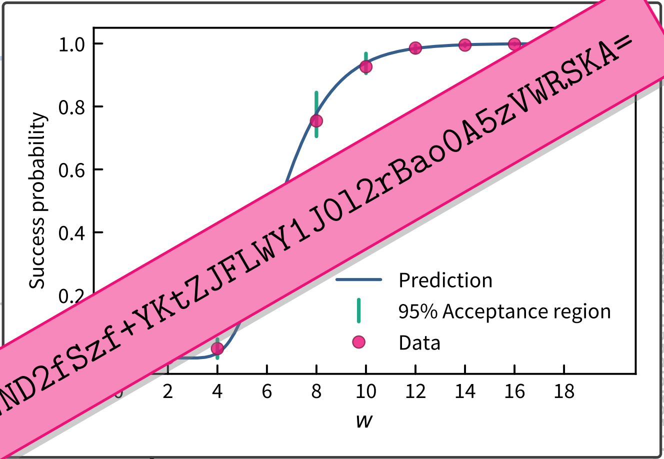


→ The support of $\mathrm{col}(\mathbf{D}')$ determines the secret.

[BCJ23] Bremner, Cheng, Ji, arXiv:2308.07152.

The radical attack





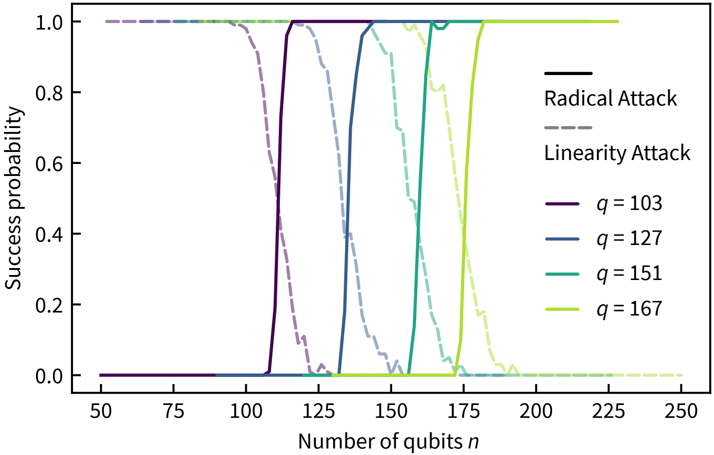
→ The support of col($\mathbf{D}'$) determines the secret.

The SECRET is cyCxfXKxLxXu3YWND2fSzf+YKtZJFLWY1J012rBao0A5zVWRSKA=

…023] Bremner, Cheng, Ji, arXiv:2308.07152.

More attacks

→ **Valleys of opportunity!**

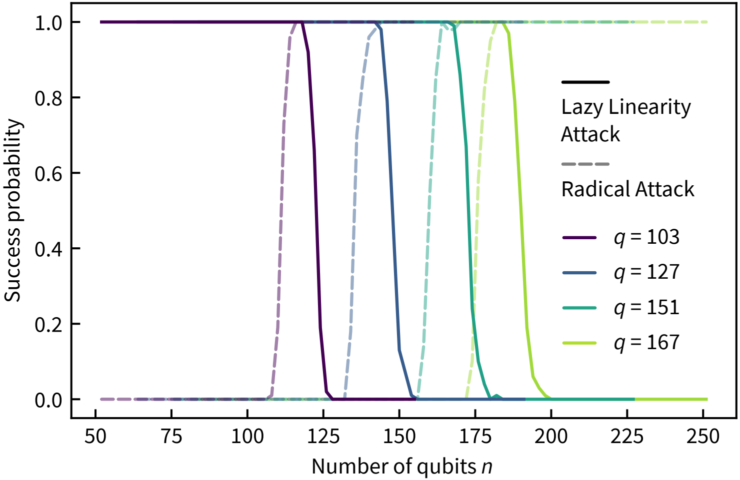


More attacks

→ The **Lazy Meyer Attack:** Only search small kernels

More attacks

→ The **Lazy Meyer Attack:** Only search small kernels

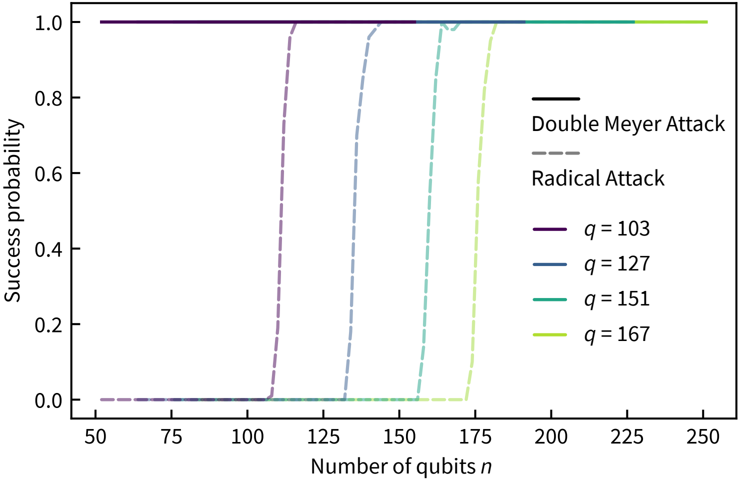


More attacks

- The **Lazy Meyer Attack:** Only search small kernels
- The **Double Meyer Attack:** Take kernel intersections to make the search space smaller

More attacks

- The **Lazy Meyer Attack:** Only search small kernels
- The **Double Meyer Attack:** Take kernel intersections to make the search space smaller



More attacks

- The **Lazy Meyer Attack:** Only search small kernels
- The **Double Meyer Attack:** Take kernel intersections to make the search space smaller
- **Hamming's razor:** identify redundant rows by exploiting that there are no low-weight Hamming strings in the image of the secret space.

THE END

Outlook

Hiding secrets

- Can large Fourier coefficients of IQP be efficiently estimated?
- Nonlinear tests?
- Can we apply similar ideas to universal circuits?
- Can we hide peaks in the **output distribution** of a circuit?
  [Aaronson-Zhang-24]

Outlook

Hiding secrets

- Can large Fourier coefficients of IQP be efficiently estimated?
- Nonlinear tests?
- Can we apply similar ideas to universal circuits?
- Can we hide peaks in the **output distribution** of a circuit? [Aaronson-Zhang-24]

Outlook

Hiding secrets

- Can large Fourier coefficients of IQP be efficiently estimated?
- Nonlinear tests?
- Can we apply similar ideas to universal circuits?
- Can we hide peaks in the **output distribution** of a circuit?
  [Aaronson-Zhang-24]

Outlook

Hiding secrets

- → Can large Fourier coefficients of IQP be efficiently estimated?
- → Nonlinear tests?
- → Can we apply similar ideas to universal circuits?
- → Can we hide peaks in the **output distribution** of a circuit?
  [Aaronson-Zhang-24]

Using interaction

- → Are there less structured interactive schemes?
- → E.g. mid-circuit measurements in a random circuit with a little bit of structure?