A one-query lower bound for unitary synthesis and breaking quantum cryptography

Fermi Ma
(Simons and Berkeley)

joint work with Alex Lombardi and John Wright

Normally, we study computational problems that can be efficiently reduced to computing some **function** $f: \{0,1\}^* \to \{0,1\}^*$.

Normally, we study computational problems that can be efficiently reduced to computing some **function** $f: \{0,1\}^* \to \{0,1\}^*$.

- **3SAT:** given a formula ϕ, compute the bit $f(\phi)$ indicating whether ϕ is satisfiable.

Normally, we study computational problems that can be efficiently reduced to computing some **function** $f: \{0,1\}^* \to \{0,1\}^*$.

- **3SAT:** given a formula ϕ, compute the bit $f(\phi)$ indicating whether ϕ is satisfiable.
- **Factoring:** given a positive integer N, compute $f(N) =$ prime factorization of N.

Normally, we study computational problems that can be efficiently reduced to computing some **function** $f: \{0,1\}^* \rightarrow \{0,1\}^*$.

- **3SAT:** given a formula ϕ, compute the bit $f(\phi)$ indicating whether ϕ is satisfiable.
- **Factoring:** given a positive integer N, compute $f(N) =$ prime factorization of N.
- **Local Hamiltonian:** given a local Hamiltonian H, compute the bit $f(H)$ indicating whether H has a low-energy ground state.

But what about problems with **quantum** inputs and outputs?

But what about problems with **quantum** inputs and outputs?

- **State tomography:** given many copies of a quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.

But what about problems with **quantum** inputs and outputs?

- **State tomography:** given many copies of a quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.
- **Quantum error correction:** given a noisy quantum codeword $|c\rangle$, recover the original message.

But what about problems with **quantum** inputs and outputs?

- **State tomography:** given many copies of a quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.
- **Quantum error correction:** given a noisy quantum codeword $|c\rangle$, recover the original message.
- **State distinguishing:** distinguish whether a given state $|\psi\rangle$ was sampled from distribution D_0 or D_1 (promised it's possible).

But what about problems with **quantum** inputs and outputs?

- **State tomography:** given many copies of a quantum state $|\psi\rangle$, output a classical description of $|\psi\rangle$.
- **Quantum error correction:** given a noisy quantum codeword $|c\rangle$, recover the original message.
- **State distinguishing:** distinguish whether a given state $|\psi\rangle$ was sampled from distribution D_0 or D_1 (promised it's possible).

Physics: computing AdS/CFT map, decoding black-hole radiation

What can complexity theory say about the hardness of these **inherently quantum** problems?

Standard procedure: reduce your problem to some well-studied complexity class.

Standard procedure: reduce your problem to some well-studied complexity class.

Ex: is the problem easy given an oracle for NP? PSPACE?

Standard procedure: reduce your problem to some well-studied complexity class.

Ex: is the problem easy given an oracle for NP? PSPACE?

Issue: for some quantum problems, it's not clear how to do this!

Standard procedure: reduce your problem to some well-studied complexity class.

Ex: is the problem easy given an oracle for $\mathbf{NP}$? $\mathbf{PSPACE}$?

Issue: for some quantum problems, it's not clear how to do this!

State distinguishing: distinguish whether a given state $|\psi\rangle$ was sampled from distribution D_0 or D_1 (promised it's possible).

Standard procedure: reduce your problem to some well-studied complexity class.

Ex: is the problem easy given an oracle for $\mathbf{NP}$? $\mathbf{PSPACE}$?

Issue: for some quantum problems, it's not clear how to do this!

State distinguishing: distinguish whether a given state $|\psi\rangle$ was sampled from distribution D_0 or D_1 (promised it's possible).

Not known how to solve this using ***any*** oracle

Standard procedure: reduce your problem to some well-studied complexity class.

Ex: is the problem easy given an oracle for $\mathbf{NP}$? $\mathbf{PSPACE}$?

Issue: for some quantum problems, it's not clear how to do this!

State distinguishing: distinguish whether a given state $|\psi\rangle$ was sampled from distribution D_0 or D_1 (promised it's possible).

Not known how to solve this using ***any*** oracle, even an oracle for the halting problem!

Before we continue:

1-minute detour for quantum computing 101

Quantum computing 101

Quantum computing 101

- n-qubit state $= 2^n$-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.

Quantum computing 101

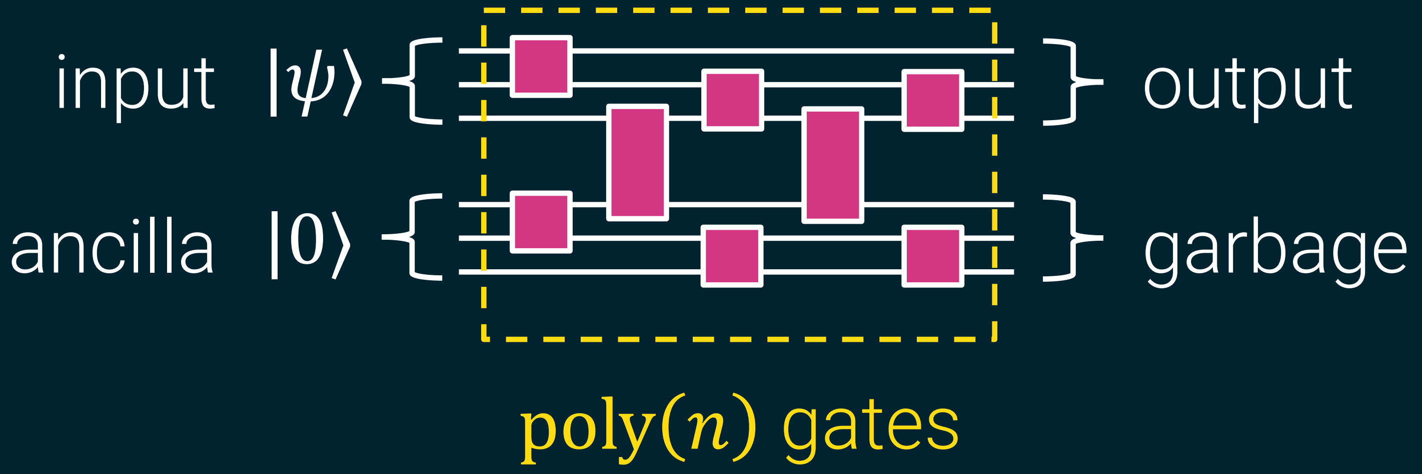
- n-qubit state = 2^n-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.
- n-qubit unitary = $2^n \times 2^n$ rotation matrix.

Quantum computing 101

- n-qubit state $=$ 2^n-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.
- n-qubit unitary $=$ $2^n \times 2^n$ rotation matrix.
- efficient quantum computation $=$ poly(n)-size quantum circuit

Quantum computing 101

- n-qubit state = 2^n-dim unit vector $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$.
- n-qubit unitary = $2^n \times 2^n$ rotation matrix.
- efficient quantum computation = poly(n)-size quantum circuit



Now back to:

Does complexity theory capture quantum problems?

Does complexity theory capture quantum problems?

Does complexity theory capture quantum problems?

- In general, a quantum problem involves computing a **unitary**.

Does complexity theory capture quantum problems?

- In general, a quantum problem involves computing a **unitary**.
- Complexity theory is about computing **functions**.

Does complexity theory capture quantum problems?

- In general, a quantum problem involves computing a **unitary**.
- Complexity theory is about computing **functions**.

To apply complexity theory, we need to ***efficiently reduce*** the task of implementing a unitary U to implementing a function f.

Does complexity theory capture quantum problems?

- In general, a quantum problem involves computing a **unitary**.
- Complexity theory is about computing **functions**.

To apply complexity theory, we need to ***efficiently reduce*** the task of implementing a unitary U to implementing a function f.

The Unitary Synthesis Problem [AK06]:

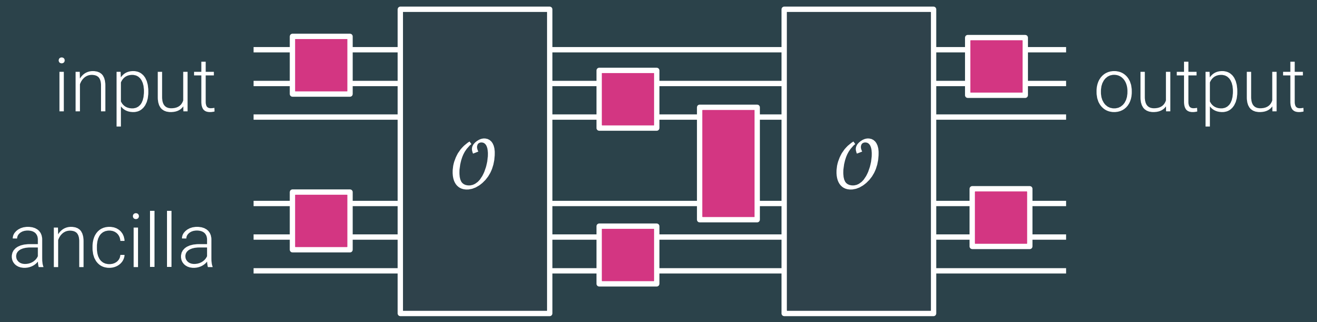
Is there a reduction for every unitary U?

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

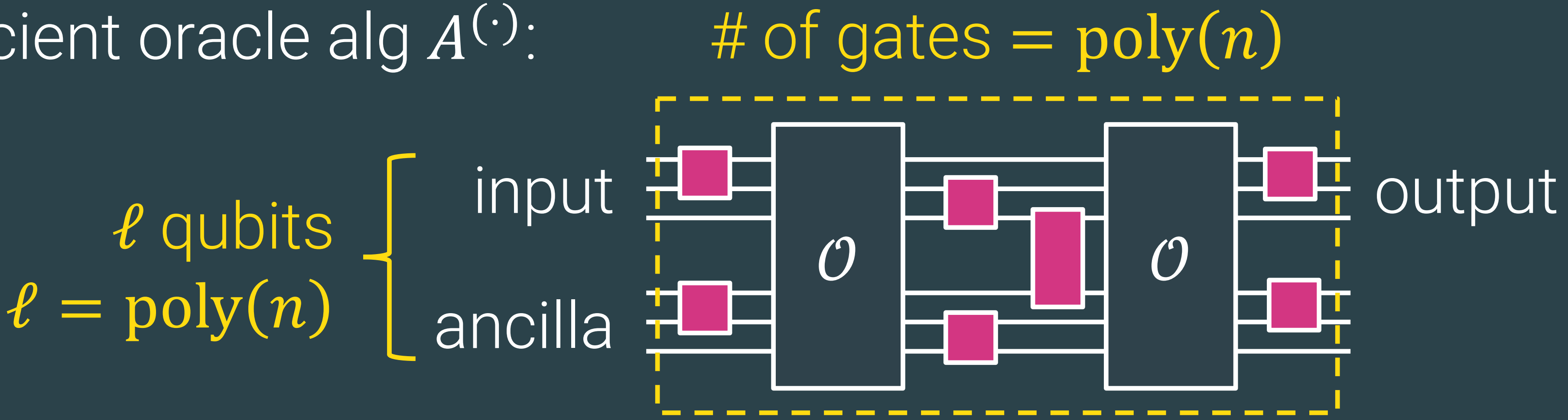
The Unitary Synthesis Problem [Aaronson-Kuperberg 06]
Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$:



The Unitary Synthesis Problem [Aaronson-Kuperberg 06]
Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$:



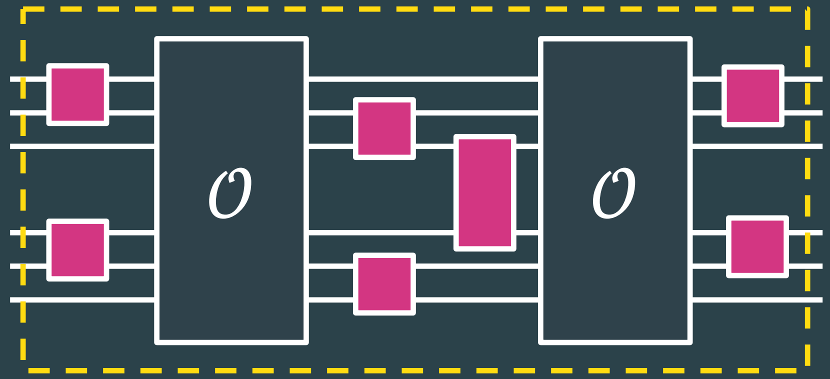
The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$: # of gates $= \mathbf{poly}(n)$

ℓ qubits
$\ell = \mathbf{poly}(n)$

input
ancilla



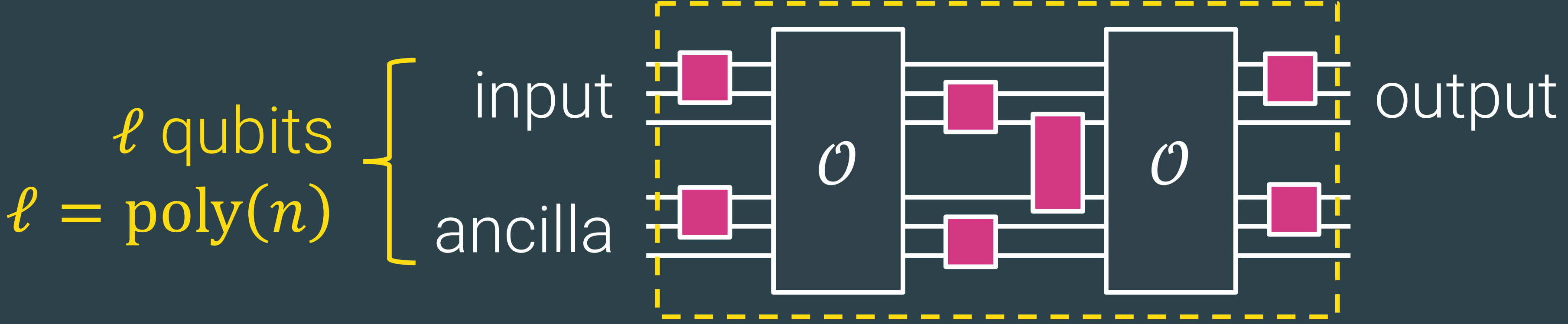
output

2) Given U, pick $f: \{0,1\}^\ell \to \{\pm 1\}$.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$: # of gates $= \mathbf{poly}(n)$



2) Given U, pick $f: \{0,1\}^\ell \to \{\pm 1\}$. Plug in $O_f : |z\rangle \to f(z) \cdot |z\rangle$.

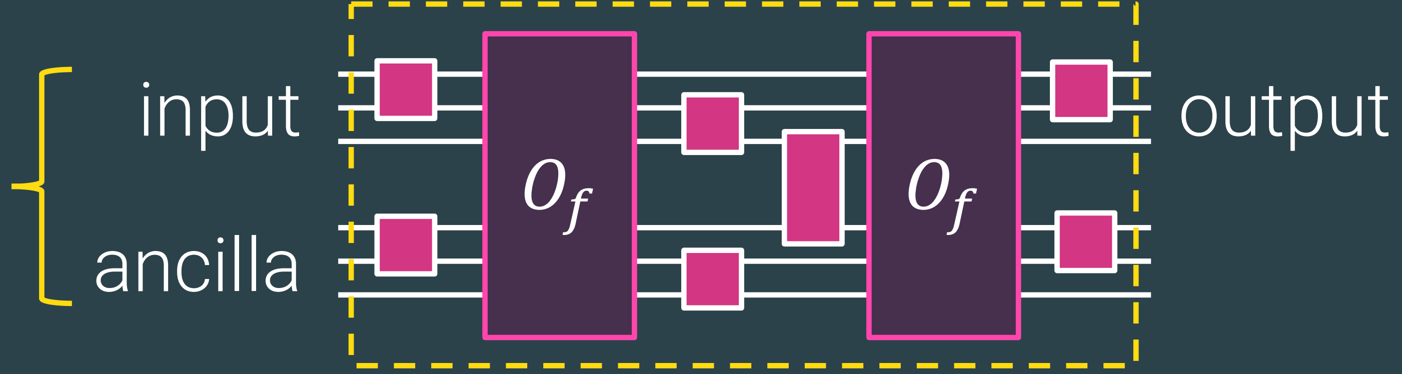
The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$: # of gates $= \mathbf{poly}(n)$

ℓ qubits

$\ell = \mathrm{poly}(n)$

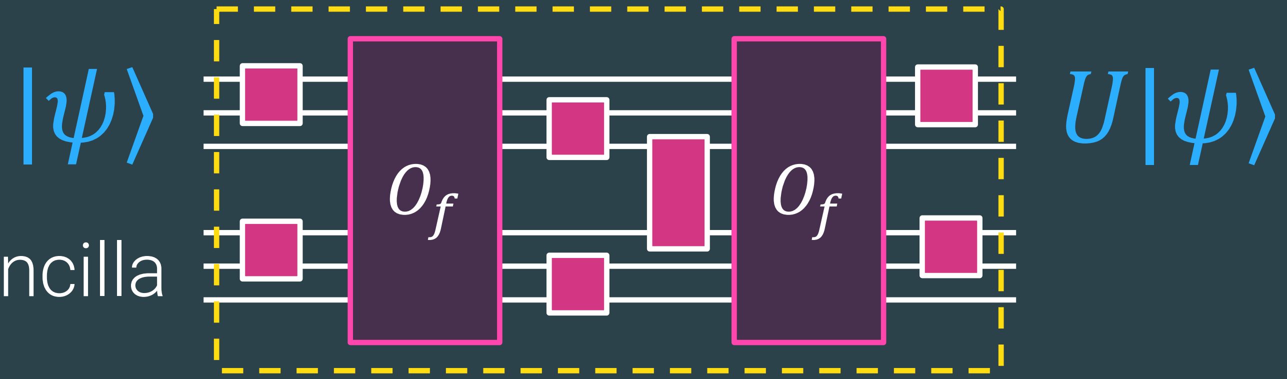


2) Given U, pick $f: \{0,1\}^\ell \to \{\pm 1\}$. Plug in $O_f: |z\rangle \to f(z) \cdot |z\rangle$.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]
Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

1) Efficient oracle alg $A^{(\cdot)}$: # of gates $= \mathbf{poly}(n)$

We want:



2) Given U, pick $f: \{0,1\}^\ell \to \{\pm 1\}$. Plug in $O_f: |z\rangle \to f(z) \cdot |z\rangle$.

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

Prior best-known bounds

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

Prior best-known bounds

- Upper bound: $2^{n/2}$ queries [Ros22]

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

Prior best-known bounds

- Upper bound: $2^{n/2}$ queries [Ros22]
- Lower bound: none

The Unitary Synthesis Problem [Aaronson-Kuperberg 06]

Is there an efficient oracle algorithm $A^{(\cdot)}$ that can implement **any** n-qubit unitary U given **some** function f?

Prior best-known bounds

- Upper bound: $2^{n/2}$ queries [Ros22]
- Lower bound: none

Note: [AK06] prove a 1-query lower bound for a very special class of oracle algorithms.

Why has it been hard to prove lower bounds?

Why has it been hard to prove lower bounds?

(1) Counting arguments don’t work.

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f:\{0,1\}^{\ell} \to \{\pm 1\}$.

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f: \{0,1\}^{\ell} \to \{\pm 1\}$.

Useless for $\ell > 2n$.

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f: \{0,1\}^{\ell} \to \{\pm 1\}$.

Useless for $\ell > 2n$.

(2) Even one-query algorithms are very powerful!

Why has it been hard to prove lower bounds?

(1) Counting arguments don't work.

- $2^{2^{2n}}$ different n-qubit unitaries (roughly).
- $2^{2^{\ell}}$ different functions $f\colon\{0,1\}^{\ell} \to \{\pm 1\}$.

Useless for $\ell > 2n$.

(2) Even one-query algorithms are very powerful!

In fact, they can solve any **classical input**, **quantum output** problem.

[Aar16, INNRY22, Ros23]

This work

Main result: There's no efficient one-query oracle algorithm for the Unitary Synthesis Problem.

This work

Main result: There's no efficient one-query oracle algorithm for the Unitary Synthesis Problem.

Actually, we even rule out computationally unbounded algorithms, as long as they query $f: \{0,1\}^{\ell} \to \{\pm 1\}$ on inputs of length $\ell = o(2^n)$.

This work

Main result: There's no efficient one-query oracle algorithm for the Unitary Synthesis Problem.

Actually, we even rule out computationally unbounded algorithms, as long as they query $f: \{0,1\}^\ell \to \{\pm 1\}$ on inputs of length $\ell = o(2^n)$.

Note: when $\ell = 2^{2n}$, possible to learn description of U in one query.

Rest of this talk

Part 1:
Connect unitary synthesis to breaking quantum cryptography

Part 2:
A special case of our proof

Rest of this talk

Part 1:
Connect unitary synthesis to breaking quantum cryptography

Part 2:
A special case of our proof

We prove our result by studying **pseudorandom states (PRS)**.

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

PRS → quantum commitments, multi-party computation, and more

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

PRS → quantum commitments, multi-party computation, and more

Big question: how hard is it to break a PRS?

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

PRS → quantum commitments, multi-party computation, and more

Big question: how hard is it to break a PRS?

Our answer: possibly harder than computing any function!

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

Result #2: Exists a PRS secure against **any efficient adversary $A^{(\cdot)}$ that queries an arbitrary function f once**

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

Result #2: Exists a PRS secure against **any efficient adversary $A^{(\cdot)}$ that queries an arbitrary function f once**, relative to a random oracle R (where f can be chosen based on R).

We prove our result by studying **pseudorandom states (PRS)**.

PRS: family of n-qubit states $\{|\mathrm{PRS}_k\rangle\}_{k\in[K]}$ where $K \ll N = 2^n$, s.t. no efficient adversary can distinguish:

- $|\mathrm{PRS}_k\rangle$ for uniformly random $k \leftarrow [K]$
- Haar-random n-qubit state $|\psi\rangle$

Result #2: Exists a PRS secure against **any efficient adversary $A^{(\cdot)}$ that queries an arbitrary function f once**, relative to a random oracle R (where f can be chosen based on R).

Note: this result implies our unitary synthesis lower bound.

Our PRS construction

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the **binary phase state**

$$|\psi_h\rangle \propto \sum_{x \in [N]} h(x)|x\rangle \qquad \text{(recall } N = 2^n\text{)}$$

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the **binary phase state**

$$|\psi_h\rangle \propto \sum_{x \in [N]} h(x)|x\rangle \qquad \text{(recall } N = 2^n\text{)}$$

Our PRS: $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where each R_k is a random function.

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the **binary phase state**

$$|\psi_h\rangle \propto \sum_{x \in [N]} h(x)|x\rangle \qquad \text{(recall } N = 2^n\text{)}$$

Our PRS: $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where each R_k is a random function.

Adversary tries to distinguish

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the **binary phase state**

$$|\psi_h\rangle \propto \sum_{x \in [N]} h(x)|x\rangle \qquad \text{(recall } N = 2^n\text{)}$$

Our PRS: $\{|\psi_{R_k}\rangle\}_{k\in[K]}$ where each R_k is a random function.

Adversary tries to distinguish

- $|\psi_{R_k}\rangle$ for random $k \leftarrow [K]$

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the **binary phase state**

$$|\psi_h\rangle \propto \sum_{x \in [N]} h(x)|x\rangle \qquad \text{(recall } N = 2^n\text{)}$$

Our PRS: $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where each R_k is a random function.

Adversary tries to distinguish

- $|\psi_{R_k}\rangle$ for random $k \leftarrow [K]$
- $|\psi_h\rangle$ for random $h: [N] \to \{\pm 1\}$

Our PRS construction

For any function $h: [N] \to \{\pm 1\}$, define the **binary phase state**

$$|\psi_h\rangle \propto \sum_{x \in [N]} h(x)|x\rangle \qquad \text{(recall } N = 2^n\text{)}$$

Our PRS: $\{|\psi_{R_k}\rangle\}_{k \in [K]}$ where each R_k is a random function.

Adversary tries to distinguish

- $|\psi_{R_k}\rangle$ for random $k \leftarrow [K]$
- $|\psi_h\rangle$ for random $h: [N] \to \{\pm 1\}$

given one query to a function f, which can depend on $R := \{R_k\}$.

Next up: what does a one-query adversary look like?

One-query adversaries

input $|\psi\rangle$

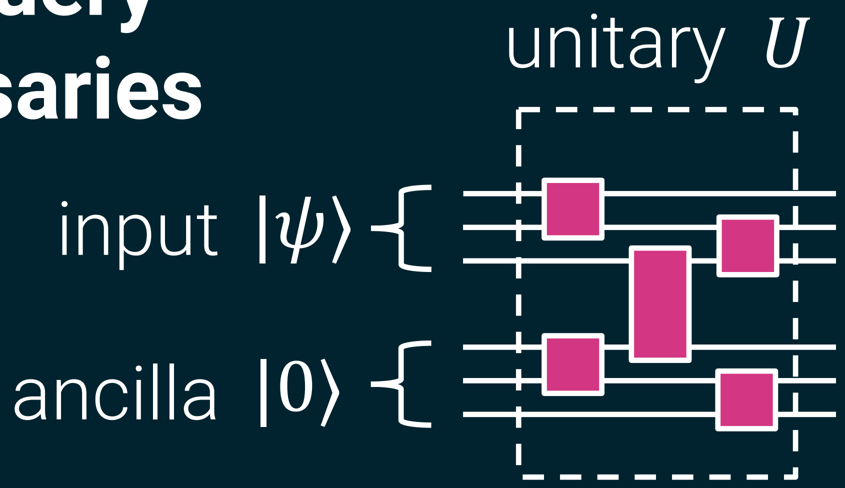
One-query adversaries

input $|\psi\rangle$

ancilla $|0\rangle$

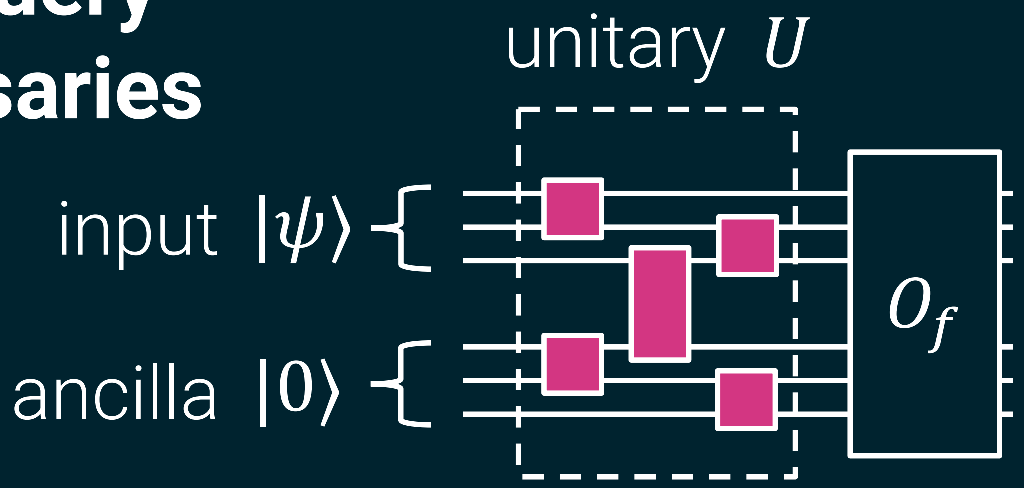
1) Initialize $\ell - n$ ancilla qubits

One-query adversaries



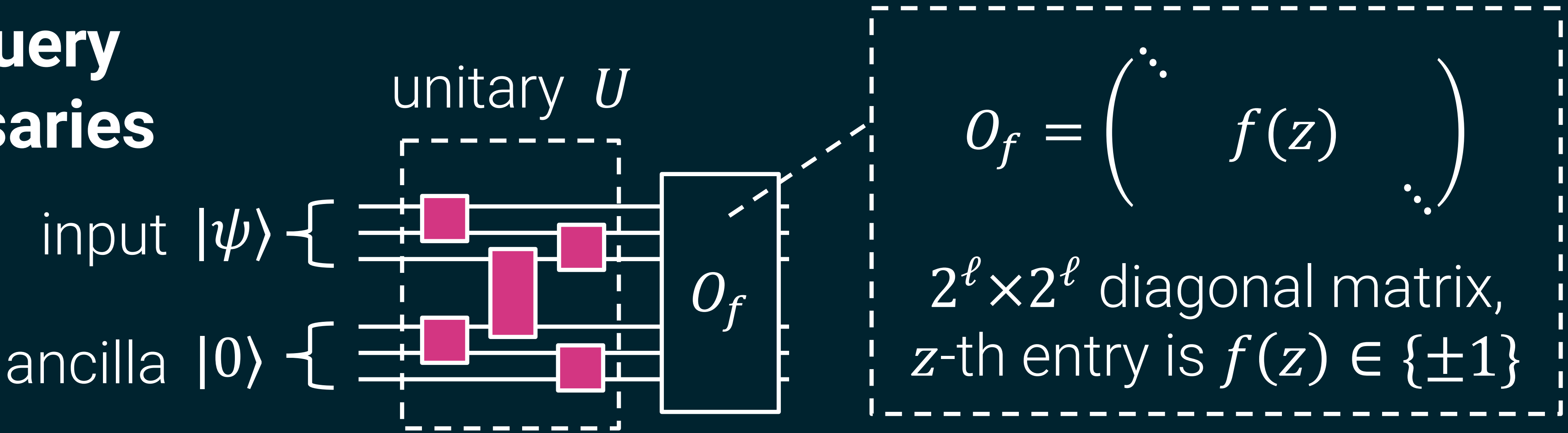
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.

One-query adversaries



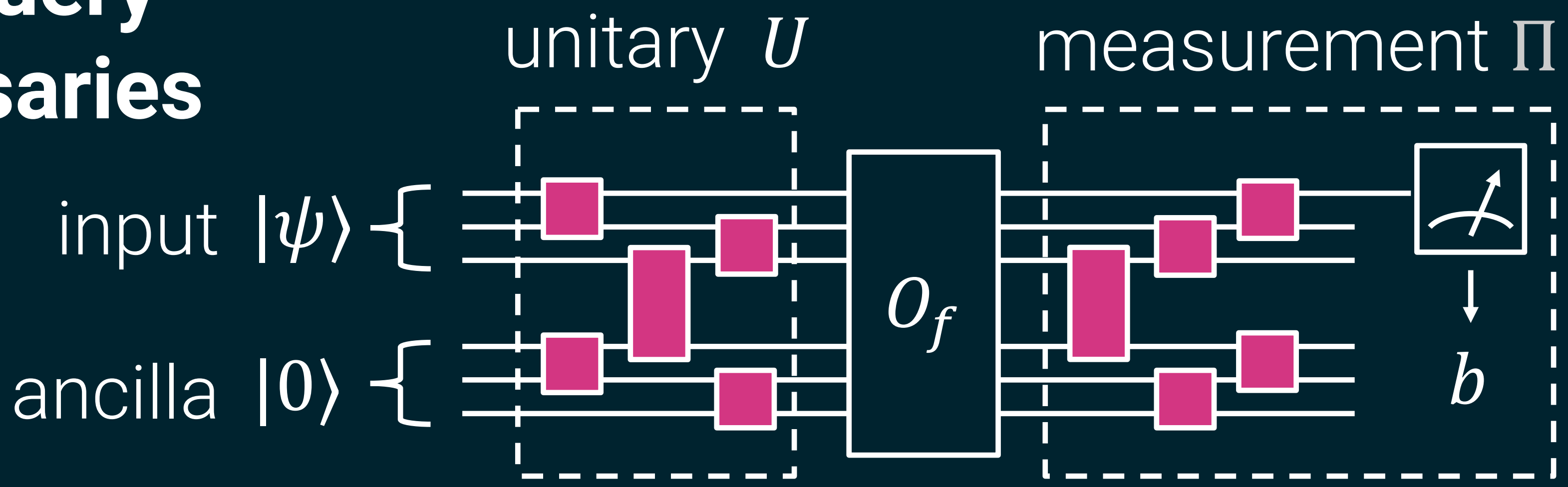
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.

One-query adversaries



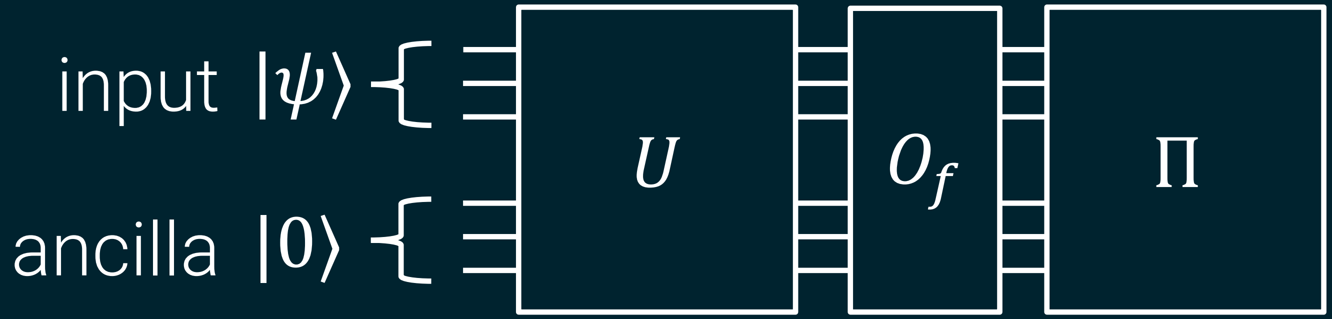
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.

One-query adversaries



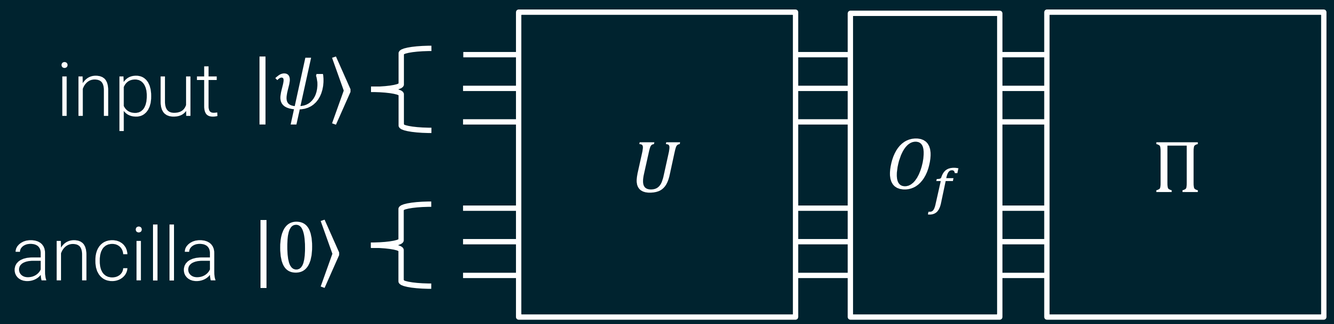
1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.
4) Measure $\{\Pi, \mathrm{I} - \Pi\}$ and return 1 if outcome is Π.

One-query adversaries



1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.
4) Measure $\{\Pi, \mathrm{I} - \Pi\}$ and return 1 if outcome is Π.

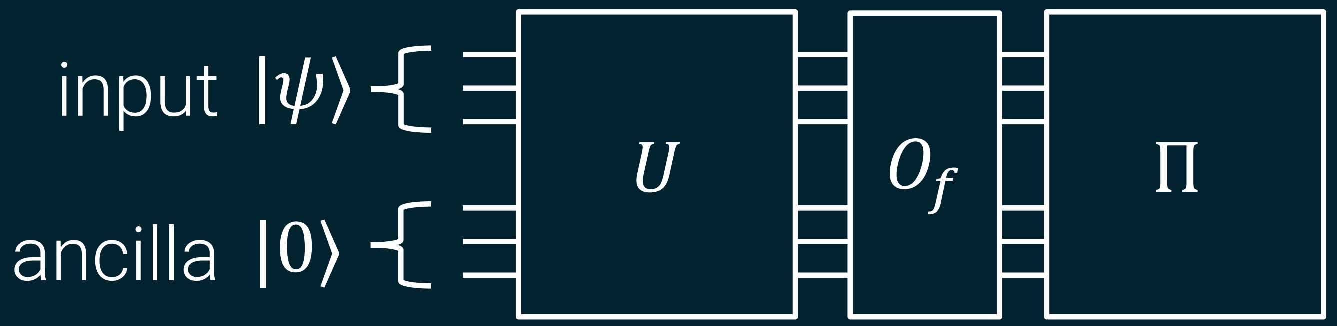
One-query adversaries



$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot U \cdot |\psi\rangle|0\rangle \right\|^2$$

1) Initialize $\ell - n$ ancilla qubits
2) Apply ℓ-qubit unitary U.
3) Query oracle O_f, which maps $|z\rangle \to f(z) \cdot |z\rangle$ for $z \in \{0,1\}^\ell$.
4) Measure $\{\Pi, \mathrm{I} - \Pi\}$ and return 1 if outcome is Π.

One-query adversaries

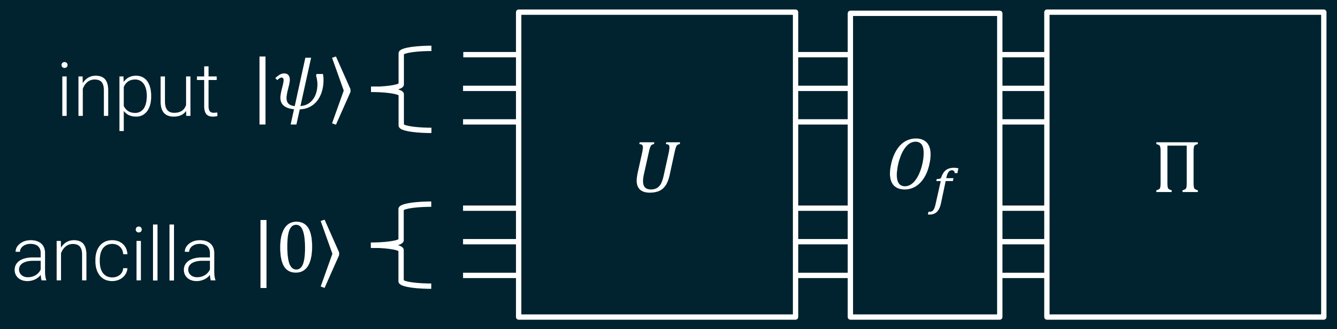


$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot U \cdot |\psi\rangle|0\rangle \right\|^2$$

Adversary's **distinguishing advantage** for fixed R is

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

One-query adversaries



$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot U \cdot |\psi\rangle|0\rangle \right\|^2$$

Adversary's **distinguishing advantage** for fixed R is

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

Goal: bound **maximum distinguishing advantage**.

Adversary's **distinguishing advantage** for fixed R is

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] \;-\; \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

Goal: bound **maximum distinguishing advantage**.

The plan:

Adversary's **distinguishing advantage** for fixed R is

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

Goal: bound **maximum distinguishing advantage**.

The plan:

1) Use spectral relaxation to bound distinguishing advantage in terms of the norm of a random matrix

Adversary's **distinguishing advantage** for fixed R is

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

Goal: bound **maximum distinguishing advantage**.

The plan:

1) Use spectral relaxation to bound distinguishing advantage in terms of the norm of a random matrix

2) Apply matrix concentration

Adversary's **distinguishing advantage** for fixed R is

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \Pr[A^f(|\psi_{R_k}\rangle) \text{ outputs } 1] - \mathop{\mathbb{E}}_{h} \Pr[A^f(|\psi_h\rangle) \text{ outputs } 1]$$

(adversary picks $f = f_R$ to maximize this)

Rest of this talk

Part 1:
Connect unitary synthesis to breaking quantum cryptography

Part 2:
A special case of our proof

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

A special class of one-query adversaries

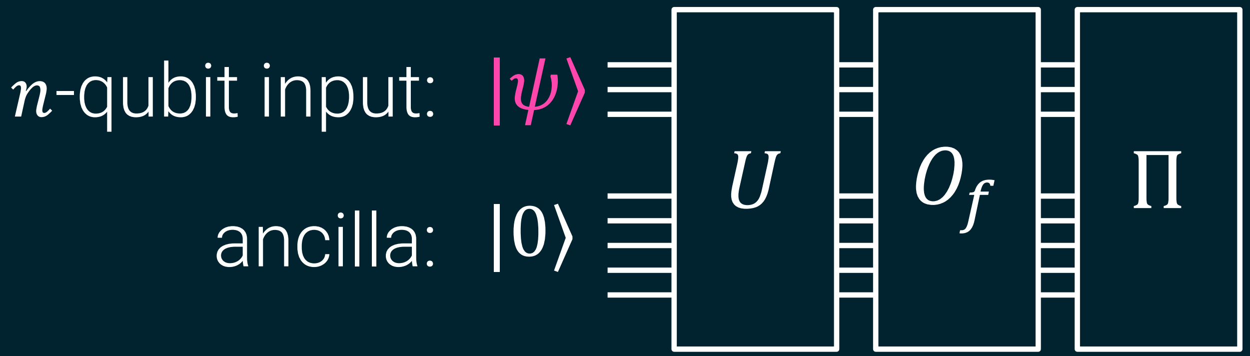
Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathbf{Id}$.

Disclaimer: We can rule out these attacks with a counting argument, but today we'll see a different proof.

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

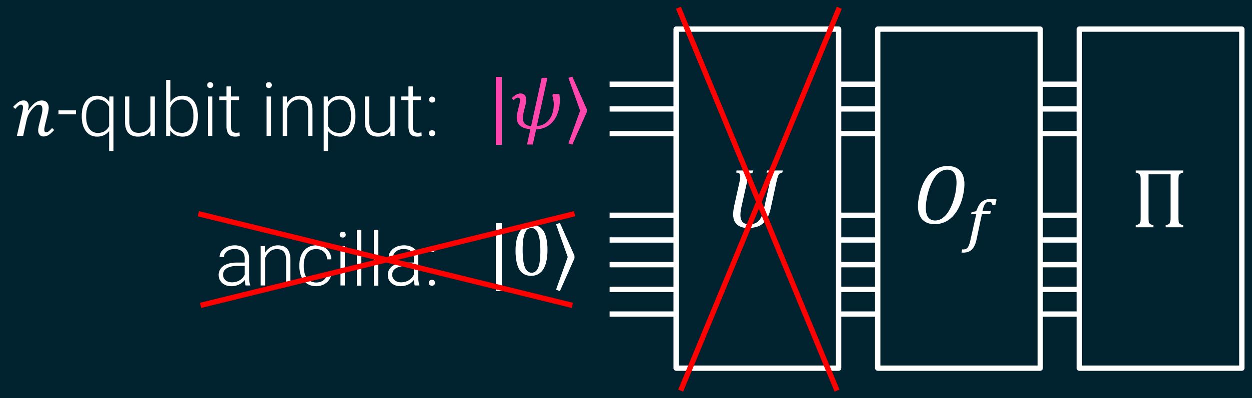
One-query adversaries:



A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

One-query adversaries:



A special class of one-query adversaries

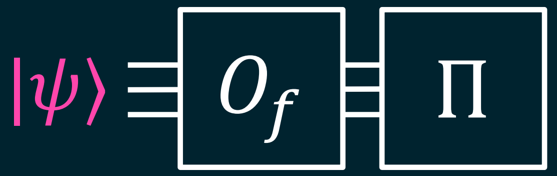
Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

Special class: n-qubit input: $|\psi\rangle \equiv \boxed{O_f} \equiv \boxed{\Pi}$

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

Special class: n-qubit input:



$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot |\psi\rangle \right\|^2$$

A special class of one-query adversaries

Assume adversary sets $\ell = n$ (no ancillas) and $U = \mathrm{Id}$.

Special class: n-qubit input: $|\psi\rangle \equiv \boxed{O_f} \equiv \boxed{\Pi}$

$$\Pr[A^f(|\psi\rangle) \text{ outputs } 1] = \left\| \Pi \cdot O_f \cdot |\psi\rangle \right\|^2$$

Distinguishing advantage:

$$\mathop{\mathbb{E}}_{k \leftarrow [K]} \langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathop{\mathbb{E}}_{h} \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle$$

(adversary picks $f = f_R$ to maximize this)

Technical tool: matrix concentration

Technical tool: matrix concentration

Scalar Chernoff bound: If X is a random **scalar** with bounded absolute value, then for i.i.d. $X_1, \dots, X_K$

$$\left|\frac{1}{K}\sum_k X_k - \mathbb{E}[X]\right| \approx O\left(\frac{1}{\sqrt{K}}\right) \qquad \text{(w.h.p.)}$$

Technical tool: matrix concentration

Scalar Chernoff bound: If X is a random **scalar** with bounded absolute value, then for i.i.d. $X_1, \dots, X_K$

$$\left| \frac{1}{K} \sum_k X_k - \mathbb{E}[X] \right| \approx O\left(\frac{1}{\sqrt{K}}\right) \qquad \text{(w.h.p.)}$$

Matrix Chernoff bound: If X is a random Hermitian $L \times L$ **matrix** with bounded operator norm, then for i.i.d. $X_1, \dots, X_K$

$$\left\| \frac{1}{K} \sum_k X_k - \mathbb{E}[X] \right\|_{\mathrm{op}} \approx O\left(\frac{\sqrt{\log(L)}}{\sqrt{K}}\right) \qquad \text{(w.h.p.)}$$

Adversary's advantage (for this special class):

$$\max_{f:[\mathrm{N}]\to\{\pm 1\}} \left| \frac{1}{K}\sum_{k} \langle\psi_{R_k}| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle\psi_h| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

Adversary's advantage (for this special class):

$$\max_{f:[\mathrm{N}]\to\{\pm 1\}} \left| \frac{1}{K}\sum_k \langle \psi_{R_k}| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle \psi_h| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

Matrix Chernoff:

$$\max_{|v\rangle} \left| \langle v| \cdot \left(\frac{1}{K}\sum_k X_k - \mathbb{E}[X] \right) \cdot |v\rangle \right|$$

Adversary's advantage (for this special class):

$$\max_{f:[N]\to\{\pm 1\}} \left| \frac{1}{K}\sum_k \langle\psi_{R_k}| \cdot \underbrace{O_f \cdot \Pi \cdot O_f}_{\text{max over matrices}} \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle\psi_h| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

random vectors

Matrix Chernoff:

$$\max_{|v\rangle} \left| \langle v| \cdot \left(\frac{1}{K}\sum_k X_k - \mathbb{E}[X] \right) \cdot |v\rangle \right|$$

random matrices

max over unit vectors

Adversary's advantage (for this special class):

$$\max_{f:[\mathrm{N}]\to\{\pm 1\}} \left| \frac{1}{K}\sum_k \langle\psi_{R_k}| \cdot \underbrace{O_f \cdot \Pi \cdot O_f}_{\text{max over matrices}} \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle\psi_h| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

random vectors

$\Downarrow$?

Matrix Chernoff:

$$\max_{|v\rangle} \left| \langle v| \cdot \left(\frac{1}{K}\sum_k X_k - \mathbb{E}[X] \right) \cdot |v\rangle \right|$$

random matrices

max over unit vectors

Adversary's advantage (for this special class):

$$\max_{f:[N]\to\{\pm 1\}} \left| \frac{1}{K}\sum_k \langle\psi_{R_k}| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle\psi_h| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

Key step: we can refactor this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$$= \frac{1}{K}\sum_k X_k - E[X]$$

f-dependent unit vector

Adversary's advantage (for this special class):

$$\max_{f:[\mathrm{N}]\to\{\pm 1\}} \left| \frac{1}{K}\sum_k \langle\psi_{R_k}| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_{R_k}\rangle - \mathbb{E}_h \langle\psi_h| \cdot O_f \cdot \Pi \cdot O_f \cdot |\psi_h\rangle \right|$$

Key step: we can refactor this as $\langle v_f| \cdot (\text{random matrix}) \cdot |v_f\rangle$

$$= \frac{1}{K}\sum_k X_k - E[X]$$

f-dependent unit vector

Then matrix Chernoff will bound the max over **all unit vectors**.

Adversary's advantage (for this special class):

$$\max_{f:[\mathrm{N}]\to\{\pm 1\}} \left| \frac{1}{K} \sum_k \boxed{\langle \psi_{R_k} | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_{R_k} \rangle} - \mathbb{E}_h \langle \psi_h | \cdot O_f \cdot \Pi \cdot O_f \cdot | \psi_h \rangle \right|$$

Since all the terms look identical, it suffices to just look at one term.

We'll rewrite this as $\langle v_f | \cdot \text{(random matrix)} \cdot | v_f \rangle$

$$\langle \psi_{R_k} |\, O_f \cdot \Pi \cdot O_f \,| \psi_{R_k} \rangle$$

We'll rewrite this as $\underbrace{\langle v_f| \cdot \text{(random matrix)} \cdot |v_f\rangle}$

$$\langle\psi_{R_k}|\, O_f \cdot \Pi \cdot O_f\, |\psi_{R_k}\rangle$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix} \cdot \frac{1}{\sqrt{N}}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

We'll rewrite this as $\langle v_f| \cdot \text{(random matrix)} \cdot |v_f\rangle$

$$\underbrace{\langle \psi_{R_k}| \, O_f \cdot \Pi \cdot O_f \, |\psi_{R_k}\rangle}$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix} \cdot \frac{1}{\sqrt{N}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$N \times N$ diagonal matrix,
x-th entry is $R_k(x)$

uniform
superposition

We'll rewrite this as $\underbrace{\langle v_f| \cdot \text{(random matrix)} \cdot |v_f\rangle}_{\langle\psi_{R_k}|\, O_f \cdot \Pi \cdot O_f\, |\psi_{R_k}\rangle}$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:= D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:= |+_N\rangle}$$

We'll rewrite this as $\langle v_f| \cdot \text{(random matrix)} \cdot |v_f\rangle$

$$\overbrace{\langle\psi_{R_k}|\, O_f \cdot \Pi \cdot O_f\, |\psi_{R_k}\rangle} = \langle +_N|\, D_{R_k} \cdot O_f \cdot \Pi \cdot O_f \cdot D_{R_k}\, |+_N\rangle \qquad \textbf{(1)}$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:=\, D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:=\, |+_N\rangle}$$

We'll rewrite this as $\langle v_f | \cdot \text{(random matrix)} \cdot | v_f \rangle$

$$\overbrace{\langle \psi_{R_k} | \, O_f \cdot \Pi \cdot O_f \, | \psi_{R_k} \rangle} = \langle +_N | \, D_{R_k} \cdot O_f \cdot \Pi \cdot O_f \cdot D_{R_k} \, | +_N \rangle \qquad \textbf{(1)}$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:= D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:= |+_N\rangle}$$

(2) O_f is a diagonal matrix, so it ***commutes*** with $D_{R\,k}$

We'll rewrite this as $\langle v_f| \cdot \text{(random matrix)} \cdot |v_f\rangle$

$$\overbrace{\langle \psi_{R_k}| \, O_f \cdot \Pi \cdot O_f \, |\psi_{R_k}\rangle} = \langle +_N | \, D_{R_k} \cdot O_f \cdot \Pi \cdot O_f \cdot D_{R_k} \, |+_N\rangle \quad \textbf{(1)}$$

$$= \langle +_N | \, O_f \cdot (D_{R_k} \cdot \Pi \cdot D_{R_k}) \cdot O_f \, |+_N\rangle \quad \textbf{(2)}$$

(1) Write the binary phase state $|\psi_{R_k}\rangle$ as

$$|\psi_{R_k}\rangle = \underbrace{\begin{pmatrix} \ddots & & \\ & R_k(x) & \\ & & \ddots \end{pmatrix}}_{:= D_{R_k}} \cdot \underbrace{\frac{1}{\sqrt{N}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{:= |+_N\rangle}$$

(2) O_f is a diagonal matrix, so it ***commutes*** with D_{R_k}

Distinguishing advantage

$$\frac{1}{K}\sum_{k}\langle\psi_{R_k}|O_f\cdot\Pi\cdot O_f|\psi_{R_k}\rangle-\mathbb{E}_h\langle\psi_h|O_f\cdot\Pi\cdot O_f|\psi_h\rangle$$

Distinguishing advantage $\frac{1}{K}\sum_k \langle\psi_{R_k}|O_f \cdot \Pi \cdot O_f|\psi_{R_k}\rangle - \mathbb{E}_h\langle\psi_h|O_f \cdot \Pi \cdot O_f|\psi_h\rangle$

Rewrite as:

$$= \langle +_N|O_f\left(\frac{1}{K}\sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h[D_h \cdot \Pi \cdot D_h]\right)O_f|+_N\rangle$$

Distinguishing advantage $\frac{1}{K}\sum_k \langle\psi_{R_k}|O_f \cdot \Pi \cdot O_f|\psi_{R_k}\rangle - \mathbb{E}_h\langle\psi_h|O_f \cdot \Pi \cdot O_f|\psi_h\rangle$

Rewrite as:

$$= \langle +_N|O_f\left(\frac{1}{K}\sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h[D_h \cdot \Pi \cdot D_h]\right)\underbrace{O_f|+_N\rangle}_{\text{unit vector}}$$

Distinguishing advantage $\frac{1}{K}\sum_k \langle \psi_{R_k} | O_f \cdot \Pi \cdot O_f | \psi_{R_k} \rangle - \mathbb{E}_h \langle \psi_h | O_f \cdot \Pi \cdot O_f | \psi_h \rangle$

Rewrite as:

$$= \langle +_N | O_f \left(\frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right) \underbrace{O_f | +_N \rangle}_{\text{unit vector}}$$

$$\leq \left\| \frac{1}{K} \sum_k D_{R_k} \cdot \Pi \cdot D_{R_k} - \mathbb{E}_h [D_h \cdot \Pi \cdot D_h] \right\|_{\mathrm{op}}$$

Distinguishing advantage $\frac{1}{K}\sum_k \langle\psi_{R_k}|O_f\cdot\Pi\cdot O_f|\psi_{R_k}\rangle - \mathbb{E}_h\langle\psi_h|O_f\cdot\Pi\cdot O_f|\psi_h\rangle$

Rewrite as:

$$= \langle +_N|O_f\left(\frac{1}{K}\sum_k D_{R_k}\cdot\Pi\cdot D_{R_k} - \mathbb{E}_h[D_h\cdot\Pi\cdot D_h]\right)\underbrace{O_f|+_N\rangle}_{\text{unit vector}}$$

$$\leq \left\|\frac{1}{K}\sum_k D_{R_k}\cdot\Pi\cdot D_{R_k} - \mathbb{E}_h[D_h\cdot\Pi\cdot D_h]\right\|_{\mathrm{op}} \approx O\left(\sqrt{\frac{n}{K}}\right)$$

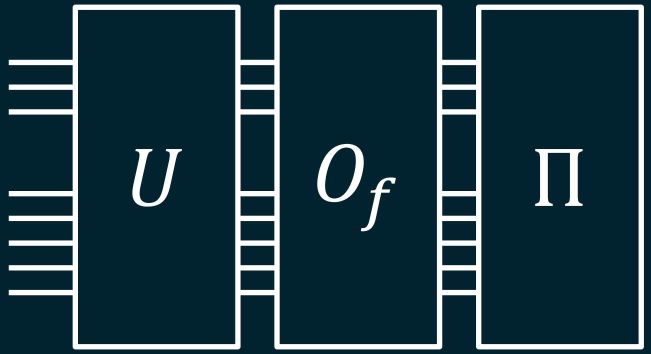
by Matrix Chernoff with $X_k = D_{R_k}\cdot\Pi\cdot D_{R_k}$

How do we handle general one-query adversaries?

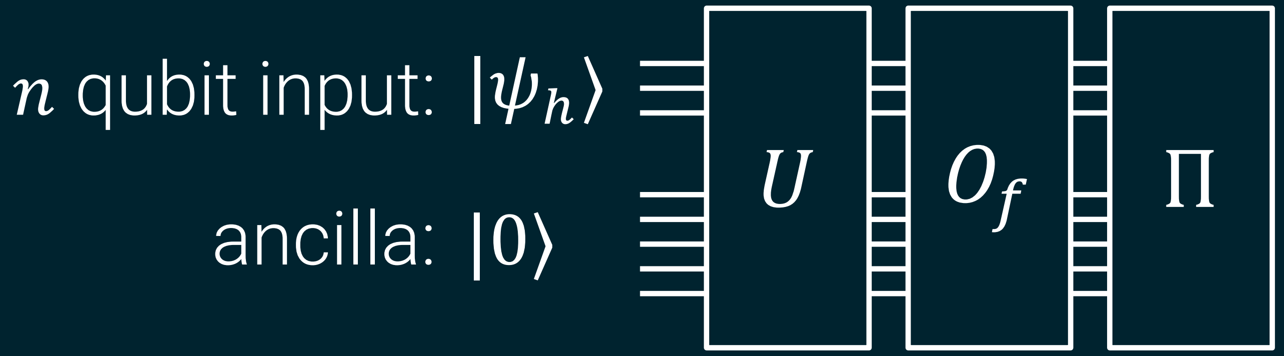
General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



General one-query adversaries

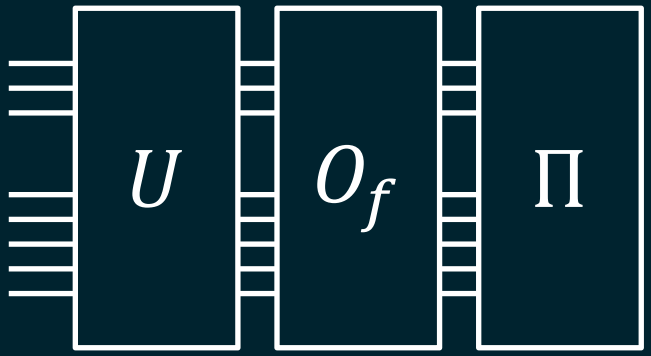


Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. “add ancillas + apply U”

General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



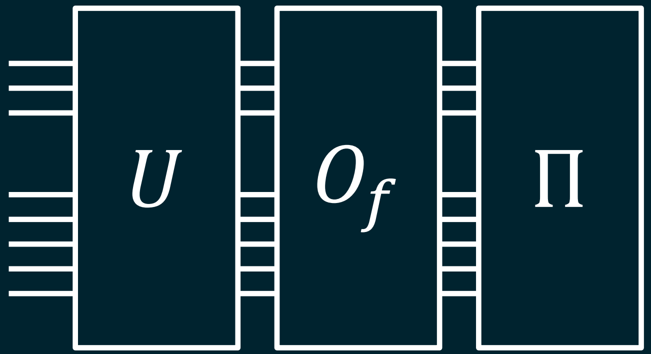
Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N|\, D_h \cdot V^\dagger \cdot O_f \cdot \Pi \cdot O_f \cdot V \cdot D_h\, |+_N\rangle$$

General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

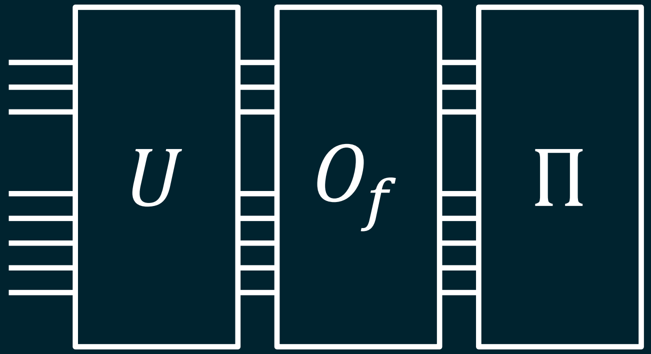
$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N |\, \underbrace{D_h \cdot V^\dagger \cdot O_f}_{} \cdot \Pi \cdot \underbrace{O_f \cdot V \cdot D_h}_{} \,| +_N \rangle$$

Challenge: unclear how to commute D_h and O_f!

General one-query adversaries

n qubit input: $|\psi_h\rangle$

ancilla: $|0\rangle$



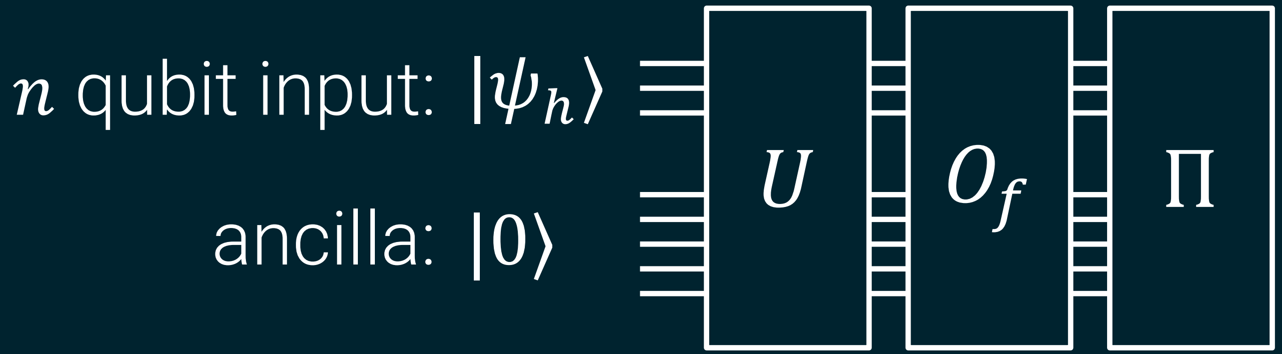
Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N|\, \underbrace{D_h \cdot V^\dagger \cdot O_f}\, \cdot \Pi \cdot \underbrace{O_f \cdot V \cdot D_h}\, |+_N\rangle$$

Challenge: unclear how to commute D_h and O_f!

Our solution: Write $V \cdot D_h\, |+_N\rangle = \widetilde{D_h}\, |\mathrm{wt}_V\rangle$ w.r.t. a V-dependent unit vector $|\mathrm{wt}_V\rangle$.

General one-query adversaries



Def: isometry $V = U \cdot (\mathrm{Id} \otimes |0\rangle)$, i.e. "add ancillas + apply U"

$$\Pr[A^f(|\psi_h\rangle) \text{ outputs } 1] = \langle +_N|\, D_h \cdot V^\dagger \cdot O_f \cdot \Pi \cdot O_f \cdot V \cdot D_h\, |+_N\rangle$$

Challenge: unclear how to commute D_h and O_f!

Our solution: Write $V \cdot D_h\, |+_N\rangle = \widetilde{D_h}\, |\mathrm{wt}_V\rangle$ w.r.t. a V-dependent unit vector $|\mathrm{wt}_V\rangle$. Commute $\widetilde{D_h}, O_f$ to get spectral relaxation.

Future directions

Future directions

Open problem #1: prove that our PRS distinguishing game is hard even given $\mathbf{poly}(n)$ queries to an arbitrary f.

Future directions

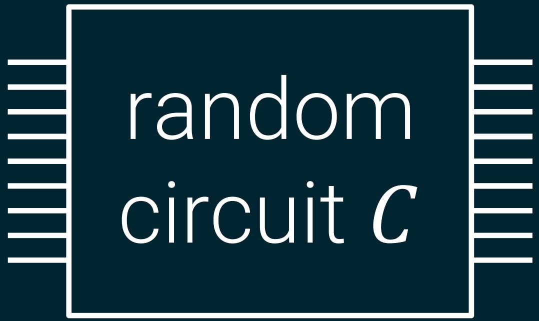
Open problem #1: prove that our PRS distinguishing game is hard even given $\mathrm{poly}(n)$ queries to an arbitrary f.

Open problem #2:

Future directions

Open problem #1: prove that our PRS distinguishing game is hard even given $\mathrm{poly}(n)$ queries to an arbitrary f.

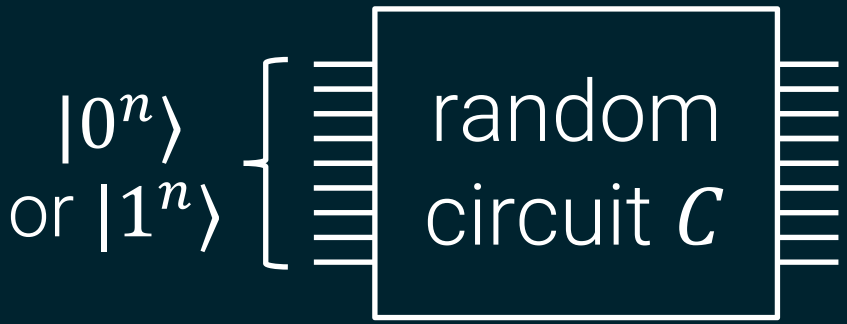
Open problem #2:



Future directions

Open problem #1: prove that our PRS distinguishing game is hard even given $\mathbf{poly}(n)$ queries to an arbitrary f.

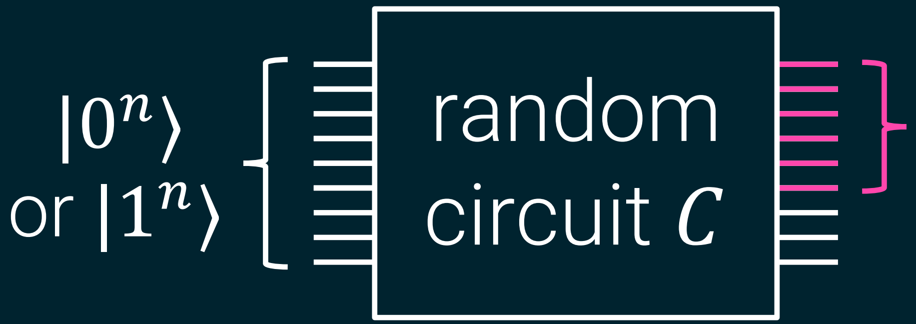
Open problem #2:



Future directions

Open problem #1: prove that our PRS distinguishing game is hard even given $\mathrm{poly}(n)$ queries to an arbitrary f.

Open problem #2:

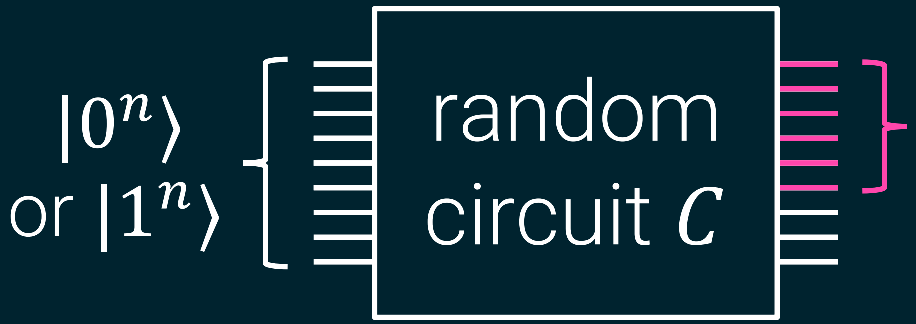


Task: given description of C and $2n/3$ qubits of $C|b^n\rangle$, determine b.

Future directions

Open problem #1: prove that our PRS distinguishing game is hard even given $\mathbf{poly}(n)$ queries to an arbitrary f.

Open problem #2:



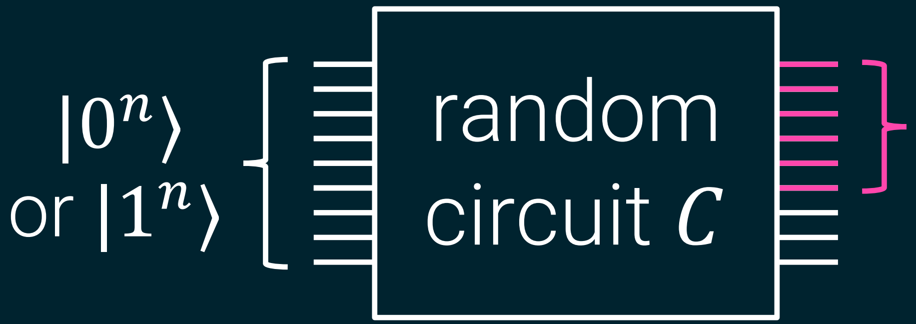
Task: given description of C and $2n/3$ qubits of $C|b^n\rangle$, determine b.

Is this easy given a halting oracle?

Future directions

Open problem #1: prove that our PRS distinguishing game is hard even given $\mathbf{poly}(n)$ queries to an arbitrary f.

Open problem #2:



Task: given description of C and $2n/3$ qubits of $C|b^n\rangle$, determine b.

Is this easy given a halting oracle?

Thanks for listening!