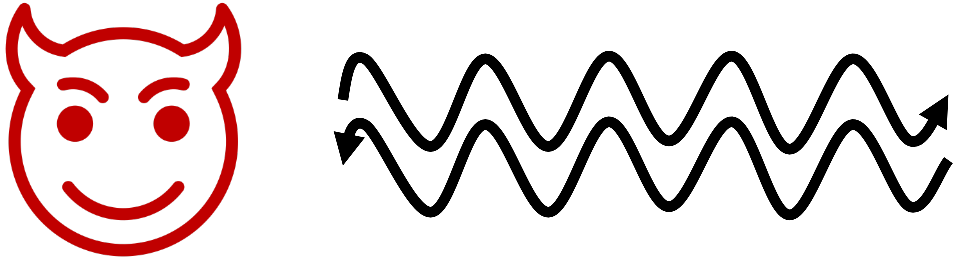
Compressed Random Oracles

Mark Zhandry (NTT Research & Stanford University)



$\mathbf{H} : \{0,1\}^m \rightarrow \{0,1\}^n$

We will always think of **H** as being a random oracle

Goal: Understand what adversary can learn about **H**

Classically "easy": Adversary knows **H(x)** for every queries point **x**, knows nothing about any other point

Note: still can be non-trivial to actually prove things

Quantumly, very hard…

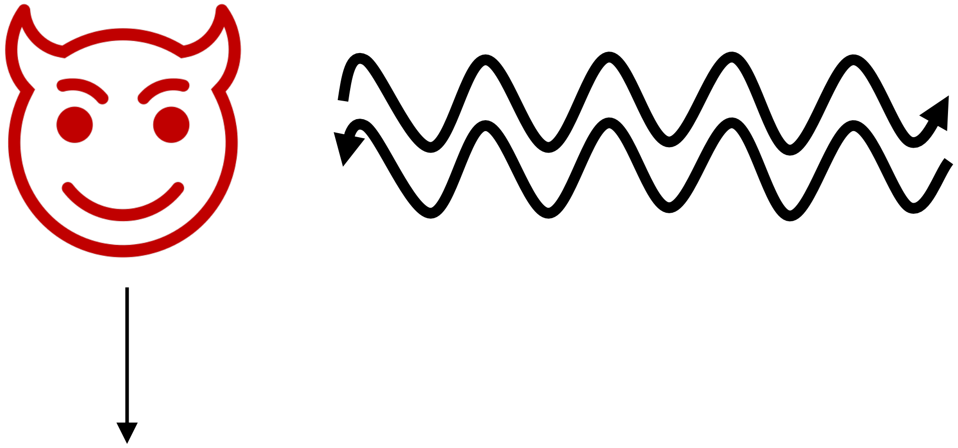
Reason: adversary “sees” all of **H** with even a single query

Usual approach: query complexity lower-bounds
= show that adversary cannot solve particular
problem with bounded queries

Examples

Lower-bound for pre-image search

[Bennett-Bernstein-Brassard-Vazirani'97]



$\mathbf{H : \{0,1\}^m \rightarrow \{0,1\}^n}$

$\mathbf{x}$ s.t. $\mathbf{H(x)=0^n}$

Lower-bound for pre-image search

[Bennett-Bernstein-Brassard-Vazirani'97]

Optimal via [Grover'96]

Thm (Adapted from [BBBV'97]): For any algorithm making **q** quantum queries to random oracle $\mathbf{H:\{0,1\}^m \rightarrow \{0,1\}^n}$ and producing an output **x**, $\mathbf{Pr[H(x)=0^n] \leq O(q^2 2^{-n})}$

Proof idea: early application of hybrid/adversary method

Start $\mathbf{H(x)\neq 0^n}$ everywhere (alg fails), then change back to original

Change adds $\mathbf{O(\sqrt{2^{-n}})}$ to root success prob for each query

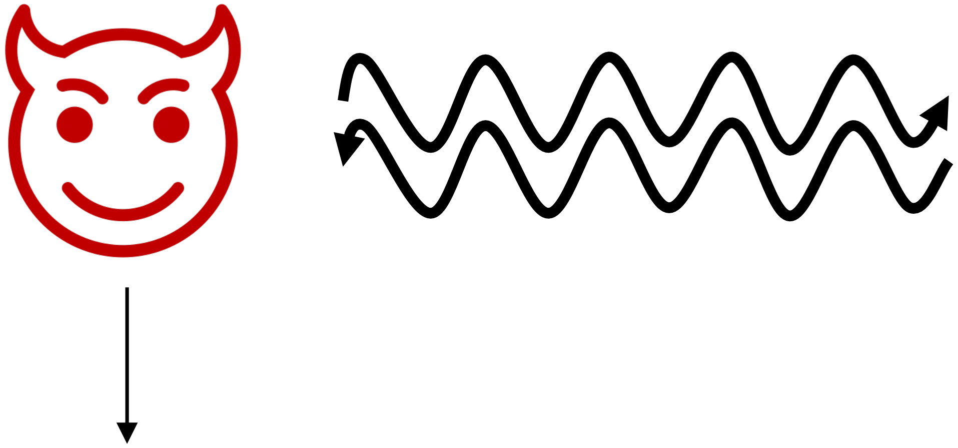
Summing and squaring gives $\mathbf{Pr[H(x)=0^n] \leq O(q^2 2^{-n})}$

Lower-bound for pre-image search

Rough intuition for quadratic speedup: Changing norm from 1-norm to 2-norm

Lower-bound for collision-finding

[Aaronson-Shi'04,...]



$\mathbf{H : \{0,1\}^m \rightarrow \{0,1\}^n}$

$\mathbf{x_0 \neq x_1}$ s.t. $\mathbf{H(x_0)=H(x_1)}$

Lower-bound for collision-finding

[Aaronson-Shi'04,...]

Optimal via [Brassard-Høyer-Tapp'98]

Thm (Adapted from [AS'97,Yuen'13,Z'15]): For any algorithm making **q** quantum queries to random oracle $\mathbf{H:\{0,1\}^m \rightarrow \{0,1\}^n}$ and producing outputs $\mathbf{x_0,x_1}$, $\mathbf{Pr[H(x_0)=H(x_1) \wedge x_0 \neq x_1] \leq O(q^3 2^{-n})}$

Proof idea: polynomial method

Observe that output probabilities are polynomials of degree **2q** in "collision parameter" (e.g. 1/number of preimages of each image)

+ show that low-degree polynomials cannot approach **0** too fast

➡ **H** indistinguishable from injective function

Q’s:

1. Why is collision-bound not $\mathbf{O(q^4/2^n)}$?

2. Do pre-image and collision bounds really need different techniques?

3. Up until 2019, all collision bounds start by showing indistinguishability from injective, which requires $\mathbf{n \geq m}$. Extending to $\mathbf{n < m}$ requires extra steps. Any “direct” proof for $\mathbf{n < m}$ case?

Compare to classical

Lower-bound for search

Lazily sample $\mathbf{H}$

$\mathbf{q}$ queries $\mathbf{x_1,\ldots x_q}$

Except with prob $\mathbf{2^{-n}}$, can assume $\mathbf{x} \in \{\mathbf{x_1,\ldots,x_q}\}$

For each $\mathbf{x_i}$, $\mathbf{Pr[H(x_i)=0^n] = 2^{-n}}$

➡ $\mathbf{Pr[\exists x_i \text{ st } H(x_i)=0^n] \leq q2^{-n}}$

Lower-bound for collision

Lazily sample $\mathbf{H}$

$\mathbf{q}$ queries $\mathbf{x_1,\ldots x_q}$

Except with prob $\mathbf{2^{-n}}$, can assume $\mathbf{x_0,x_1} \in \{\mathbf{x_1,\ldots,x_q}\}$

For $\mathbf{x_i,x_j}$, $\mathbf{Pr[H(x_i)=H(x_j)] = 2^{-n}}$

➡ $\mathbf{Pr[\exists x_i \neq x_j \text{ st } H(x_i)=H(x_j)] \leq q^2 2^{-n}}$

Compressed Oracles: An Inherently Quantum Approach

Idea: Purify **H**, extract info from purification

Usual model:

$\mathbf{H \leftarrow Funcs(\{0,1\}^m \rightarrow \{0,1\}^n)}$

Query = apply unitary $\mathbf{Q_H|x,y\rangle = |x,y \oplus H(x)\rangle}$

Purified model:

Initialize oracle register to $\mathbf{\sum_H |H\rangle}$

Query = apply unitary $\mathbf{Q|x,y,H\rangle = |x,y \oplus H(x),H\rangle}$

Idea: Purify $\mathbf{H}$, extract info from purification

Purified model:

Initialize oracle register to $\sum_{\mathbf{H}} |\mathbf{H}\rangle$

Query = apply unitary $\mathbf{Q}|\mathbf{x},\mathbf{y},\mathbf{H}\rangle = |\mathbf{x},\mathbf{y}\oplus\mathbf{H}(\mathbf{x}),\mathbf{H}\rangle$

Fourier model: view $\mathbf{y},\mathbf{H}$ in Hadamard/Fourier basis

Initialize oracle register to $|\mathbf{0}\rangle$

Query = apply unitary $\mathbf{Q}|\mathbf{x},\mathbf{z},\mathbf{J}\rangle = |\mathbf{x},\mathbf{z},\mathbf{J}\oplus\mathbf{P}_{\mathbf{x},\mathbf{z}}\rangle$

$\mathbf{P}_{\mathbf{x},\mathbf{z}}(\mathbf{x}') = \mathbf{z}$ iff $\mathbf{x}=\mathbf{x}'$

Idea: Purify **H**, extract info from purification

Fourier model: view **y,H** in Hadamard/Fourier basis

Initialize oracle register to $|\mathbf{0}\rangle$

Query = apply unitary $\mathbf{Q}|\mathbf{x,z,J}\rangle = |\mathbf{x,z,J} \oplus \mathbf{P}_{\mathbf{x,z}}\rangle$

$\mathbf{P}_{\mathbf{x,z}}(\mathbf{x'}) = \mathbf{z}$ iff $\mathbf{x=x'}$

Observation: after **q** queries, **J** register will be non-zero at only **J** points

Idea: Purify **H**, extract info from purification

Fourier model: view **y,H** in Hadamard/Fourier basis

Initialize oracle register to $|\mathbf{0}\rangle$

Query = apply unitary $\mathbf{Q}|\mathbf{x,z,J}\rangle = |\mathbf{x,z,J}\oplus\mathbf{P_{x,z}}\rangle$

Compressed Fourier model:

Initialize oracle register to $|\{\}\rangle$

Query = apply unitary $\mathbf{Q}|\mathbf{x,z,D}\rangle = |\mathbf{x,z,D}\oplus\{(\mathbf{x,z})\}\rangle$

$$\mathbf{D}\oplus\{(\mathbf{x,z})\} = \begin{cases} \mathbf{D}\setminus\{(\mathbf{x,z})\} & \text{if } (\mathbf{x,z})\in\mathbf{D} \\ (\mathbf{D}\setminus\{(\mathbf{x,z'})\}) \cup \{(\mathbf{x,z}\oplus\mathbf{z'})\} & \text{if } (\mathbf{x,z'})\in\mathbf{D},\ \mathbf{z'}\neq\mathbf{z} \\ \mathbf{D}\cup\{(\mathbf{x,z})\} & \text{if no pair } (\mathbf{x,z'})\notin\mathbf{D} \end{cases}$$

Idea: Purify $\mathbf{H}$, extract info from purification

Compressed Fourier model:

Initialize oracle register to $|\{\}\rangle$

Query = apply unitary $\mathbf{Q}|x,z,D\rangle = |x,z,D\oplus\{(x,z)\}\rangle$

$$\mathbf{D\oplus\{(x,z)\}} = \begin{cases} \mathbf{D\setminus\{(x,z)\}} & \text{if } \mathbf{(x,z)\in D} \\ \mathbf{(D\setminus\{(x,z')\})} \cup \mathbf{\{(x,y\oplus y')\}} & \text{if } \mathbf{(x,z')\in D,\ z'\neq z} \\ \mathbf{D\cup\{(x,z)\}} & \text{if no pair } \mathbf{(x,z')\notin D} \end{cases}$$

Compressed *Standard* model: move $\mathbf{z}$ registers back to primal values $\mathbf{y}$…

Joint state of adversary and oracle is pure → oracle register contains all information about adversary

Oracle "database" **D** looks a lot like list of query input/output pairs (i.e. what a classical-query adversary would know about **H**)

Key differences:

- **D(x)** not quite equal to adversary's understanding of **H(x)** since support is orthogonal to $\sum_y |y\rangle$
- Always ready to remove record from database to indicate forgetting info

Key Lemma [Z'19]: Let **R** be some relation on tuples of pairs **$(x_1,y_1),\ldots,(x_k,y_k)$**. Let **A** be a quantum query algorithm. Define

- **p** = **Pr[A** outputs a tuple in **R** such that **$y_i = H(x_i)$** for all **I**
- **p' = Pr[**compressed standard database **D** contains tuple in **R]**

Then $\sqrt{p} \le \sqrt{p'} + \sqrt{(k2^{-n})}$

Examples:

- If **A** can find pre-image of **0**, then **D** must contain **(x,0)** pair
- If **A** can find collision, then **D** must contain pairs **$(x_0,y),(x_1,y)$** with **$x_0 \neq x_1$**

Lemma [Z'19]: After **q** queries, oracle's state supported on **D** of size at most **q**

In particular, compressed oracle gives a way to lazily sample random oracles

Lemma [Z'19]: Let $\mathbf{t_i}$ be amplitude (root probability) after **i** queries on databases containing a $\mathbf{(x,0^n)}$ pair. Then $\mathbf{t_{i+1} \leq t_i + O(\sqrt{2^{-n}})}$

Corollary (reproving [BBBV'97]): For any algorithm making **q** quantum queries to random oracle $\mathbf{H:\{0,1\}^m \rightarrow \{0,1\}^n}$ and producing an output **x**, $\mathbf{Pr[H(x)=0^n] \leq O(q^2 2^{-n})}$

Proof: $\mathbf{t_0 = 0}$. By Lemma, $\mathbf{t_q \leq O(q\sqrt{2^{-n}})}$.

Then by Key Lemma, $\mathbf{\sqrt{p} \leq O(q\sqrt{2^{-n}}) + \sqrt{(2^{-n})}} = \mathbf{O(q\sqrt{2^{-n}})}$

Thus $\mathbf{p \leq O(q^2 / 2^n)}$

Lemma [Z'19]: Let $\mathbf{t_i}$ be amplitude (root probability) after **i** queries on databases containing pairs $\mathbf{(x_0,y),(x_1,y)}$ with $\mathbf{x_0 \neq x_1}$. Then $\mathbf{t_{i+1} \leq t_i + O(\sqrt{i}\,\sqrt{2^{-n}})}$

Proof idea: Can replace $\mathbf{0^n}$ in pre-image search with Lemma with any of the current entries in **D**. Naively summing over all **≤i** such entries gives $\mathbf{O(i\,\sqrt{2^{-n}})}$. But error corresponding to each entry is orthogonal → summing error vectors gives $\mathbf{O(\sqrt{i}\,\sqrt{2^{-n}})}$

Corollary (reproving [AS'97,Yuen'13,Z'15]): For any algorithm making **q** quantum queries to random oracle $\mathbf{H:\{0,1\}^m \to \{0,1\}^n}$ and producing outputs $\mathbf{x_0,x_1}$, $\mathbf{Pr[H(x_0)=H(x_1) \wedge x_0 \neq x_1] \leq O(q^3 2^{-n})}$

Why $\mathbf{O(q^4/2^n)}$ for collision? One of the classical **q**'s gets squared due to changing norms, but the other **q** doesn't since the collisions with the existing database are orthogonal

Proofs for pre-image search and collision-finding only differ in a few lines! (though pre-image search admittedly more complex)

Proof for collision-finding directly handles small-output regime

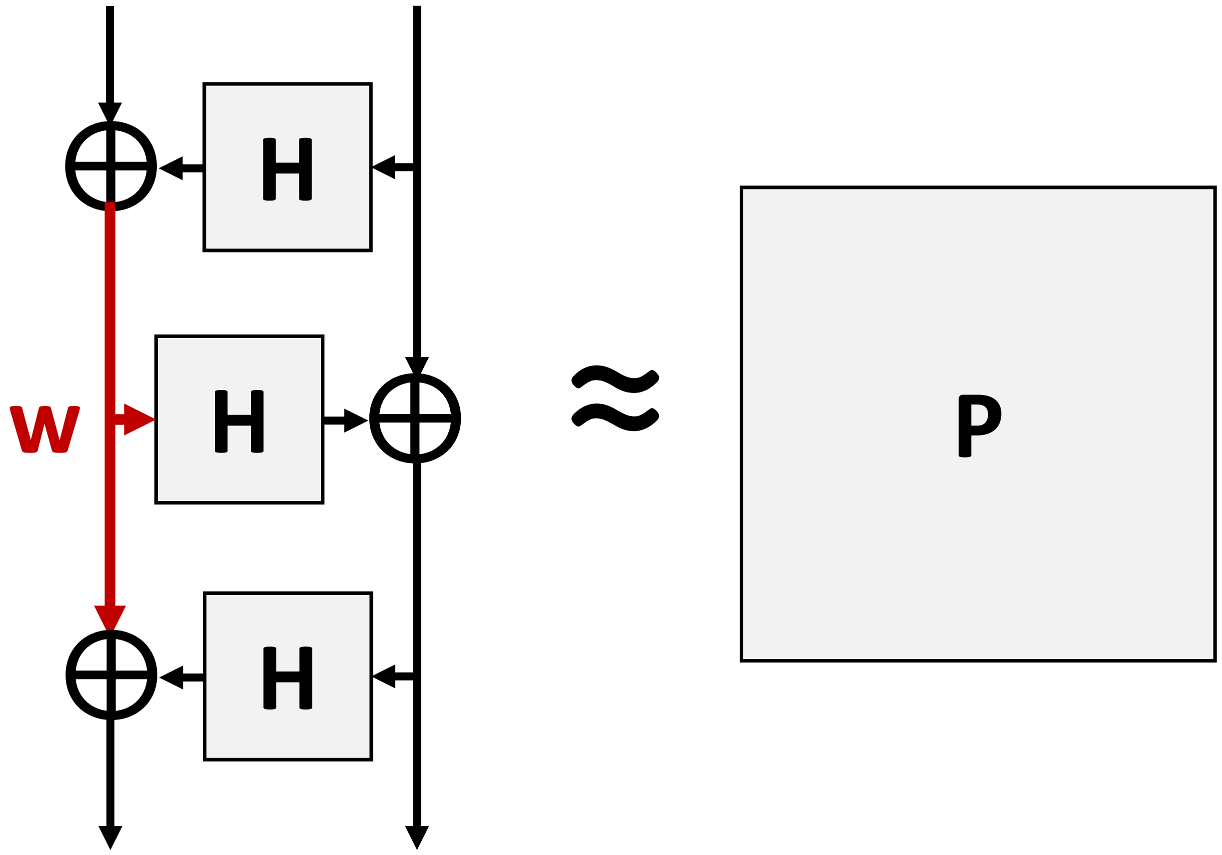
Challenge: how far can similarities to classical take us?

We know this intuition must sometimes fail…

Example: Feistel/Luby-Rackoff

[Luby-Rackoff'88]

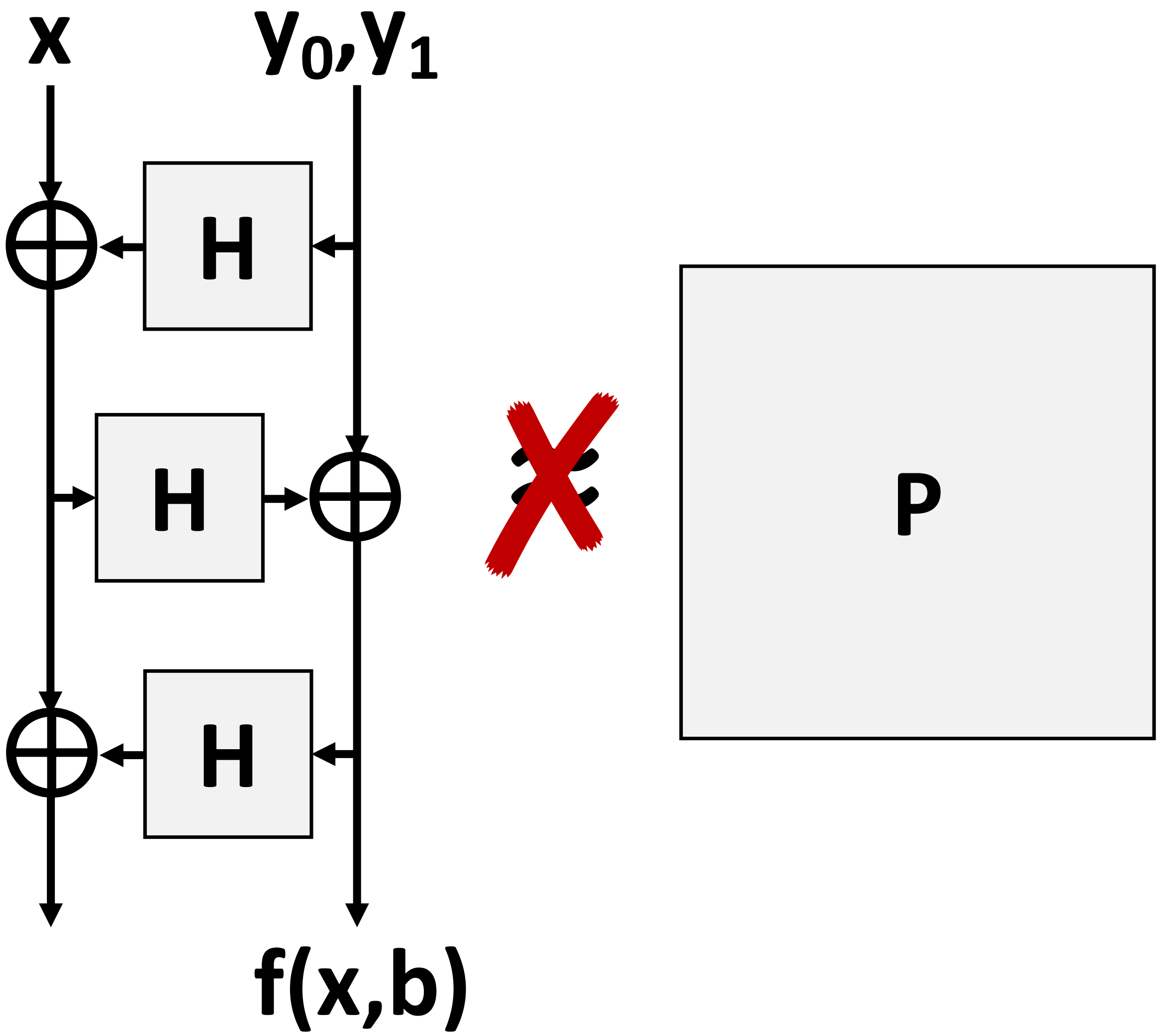
Under classical queries:



Very rough idea: look at queries to **H** on the left, show that no collisions in **w**

Example: Luby-Rackoff

[Kuwakado-Morii'10]: False under quantum queries!

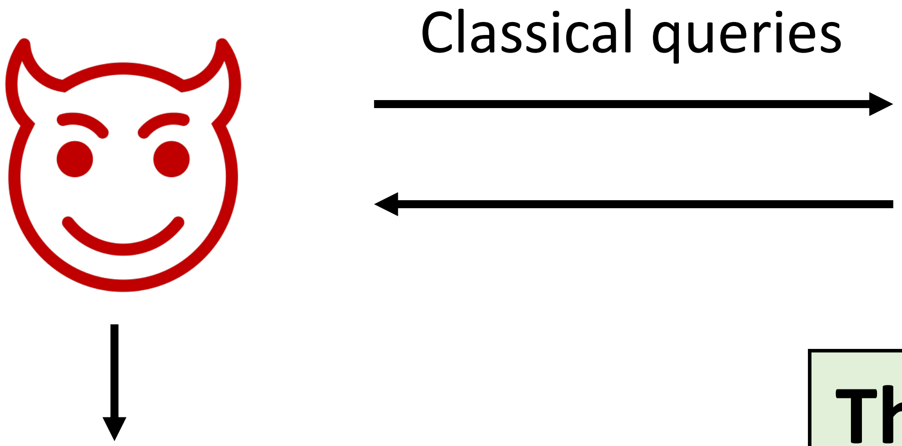


Idea:

$\mathbf{f(x,0) = f(x \oplus H(y_0) \oplus H(y_1), 1)}$

Simon's alg → $\mathbf{H(y_0) \oplus H(y_1)}$

Example: Yamakawa-Z'22



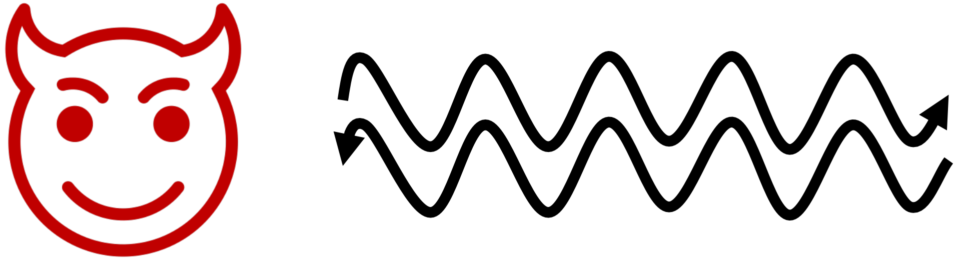
$$\mathbf{H : \{0,1\}^m \rightarrow \{0,1\}}$$

$\mathbf{x_1, x_2, \ldots, x_n}$ such that
$\mathbf{H(x_i) = 1\ \forall i}$, and
$\mathbf{(x_1, \ldots, x_n)}$ in linear code **C**

Theorem [YZ'22]: For appropriate code **C**, exponential classical queries required

Idea: **C** guarantees only a few valid solutions amongst queries made by adversary

Example: Yamakawa-Z'22



$\mathbf{H : \{0,1\}^m \rightarrow \{0,1\}}$

$\mathbf{x_1,x_2,\ldots,x_n}$ such that
$\mathbf{H(x_i) = 1}$ $\forall$**i**, and
$\mathbf{(x_1,\ldots,x_n)}$ in linear code **C**

Theorem [YZ'22]: For appropriate code **C** (consistent with classical hardness) polynomial quantum queries sufficient

In particular, a random code will do if we don't care about computation

Takeaway: if we try to classically reason about oracle database in Luby-Rackoff or Yamakawa-Z, we will get wrong answer

Variants of compressed oracles

Uniform in over other output domains

For **{0,1}**, can remove **y** register all together

Don't compress after Fourier domain

Simpler oracle, but may have to work harder to extract adversary's knowledge

Non-uniform outputs

Naively needs independence between inputs

Some open questions I would like to see answered

Q1: Better intuitive understanding of when compressed oracles work, when they don’t

Q2: Cleaner techniques for using compressed oracles

Q3: Quantum security of Feistel?

Q3a: 4-round Feistel quantum secure?

[Hosoyamada-Iwata’19, Bhaumik-Cogliati-Ethan-Jha’24]: non-adaptive queries

Q3b: 5-round secure under inverse queries?

Note: other inefficient constructions do have proven security [**Z**’16]

Q4: Online, small output time-space tradeoffs

Q4a: What is the time-space complexity of collisions?

Thm (Based on [Pollard'75, '78]): Can find collisions classically in $\mathbf{O(2^{n/2})}$ queries and space **poly(n)**

Thm [Brassard-Høyer-Tapp'98]: Can find collisions quantumly in $\mathbf{O(2^{n/3})}$ queries and space $\mathbf{2^{n/3}poly(n)}$. More generally, with space **S poly(n)**, can find collisions in $\mathbf{q \geq \Omega(2^{n/3})}$ queries, provided $\mathbf{q^2 S \geq \Omega(2^n)}$

However, known lower-bounds consistent with $\mathbf{O(2^{n/3})}$ queries and **poly(n)** space

Essentially all existing results for random oracles: either bound offline (preprocessing) space only, or consider large-output problems

Quantum Preprocessing model ([Nayebi-Aaronson-Belovs-Trevisan'15, Chung-Guo-Liu-Qian'20, Guo-Li-Liu-Zhang'21, Akshima-Guo-Liu'22,…]):

- Unbounded offline phase produces short "advice".
- Query-bounded online stage. Unlimited storage

Large output problems ([Klauck-Špalek-de Wolf'04, Hamoudi-Magniez'23]):

- Large output size **T** (e.g. find **T** collisions)
- Online storage less than **~T**

Idea for time-space lower-bound for collisions

Observation: Any purification can be compressed to adversary's storage

Idea: Since compressed oracle is purification, maybe, if adversary's storage is as most **S**, then at most **S** records in database **D**

Doesn't work: compressed oracle is not optimal-space purification

Idea for time-space lower-bound for collisions

New Idea: If we can compress database further, it means some entries are mostly un-entangled with adversary

Maybe for such entries, each query can only add $\mathbf{2^{-n}}$ to amplitude (root probability), as opposed to $\mathbf{\sqrt{2^{-n}}}$

(seems plausible, but I don't know how to prove it…)

$\Rightarrow$ Each query adds $\mathbf{O(\sqrt{S}\sqrt{2^{-n}}) + (\sqrt{q})\, 2^{-n}) = O(\sqrt{S}\sqrt{2^{-n}})}$ to amplitude

$\Rightarrow$ $\mathbf{q}$ queries adds $\mathbf{O(q\sqrt{S}\sqrt{2^{-n}})}$ to amplitude

$\Rightarrow$ Constant amplitude requires $\mathbf{q^2 S \geq \Omega(2^n)}$

Idea for time-space lower-bound for collisions

Another issue: technique only works for non-measuring collision-finders

Reason: if adversary measures, then oracle is no longer a purification

[**Z**’24]: highly structured oracle relative to which non-measuring algorithms strictly weaker than measuring ones

Q5: Building Oracles from Other Oracles

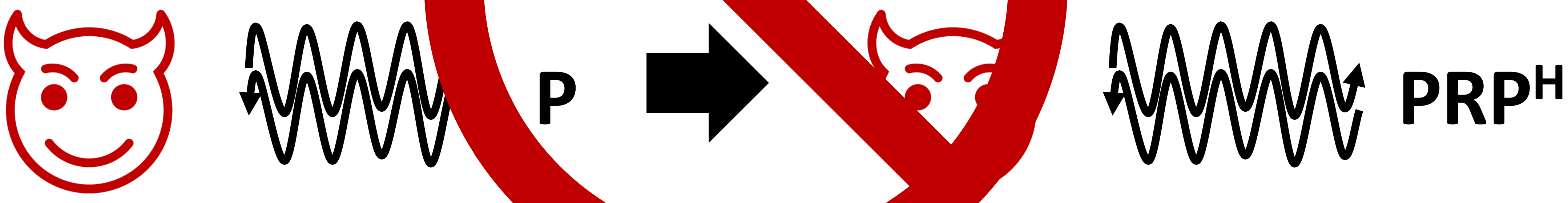
Q5a: Are ideal ciphers and random oracles equivalent?

Can you "lift" separations in one oracle model to another?

e.g. Suppose you have an oracle separation relative to a permutation oracle **P** (with inverse). Can you turn it into a separation using a random oracle **H**?

Attempt 1: Indistinguishability

Consider a P built from a ndom function H
(e.g. [Z 6]). an this lift in ossibilities?



Separation using **P** ➡ Separation using **H**???

Attempt 1: Indistinguishability

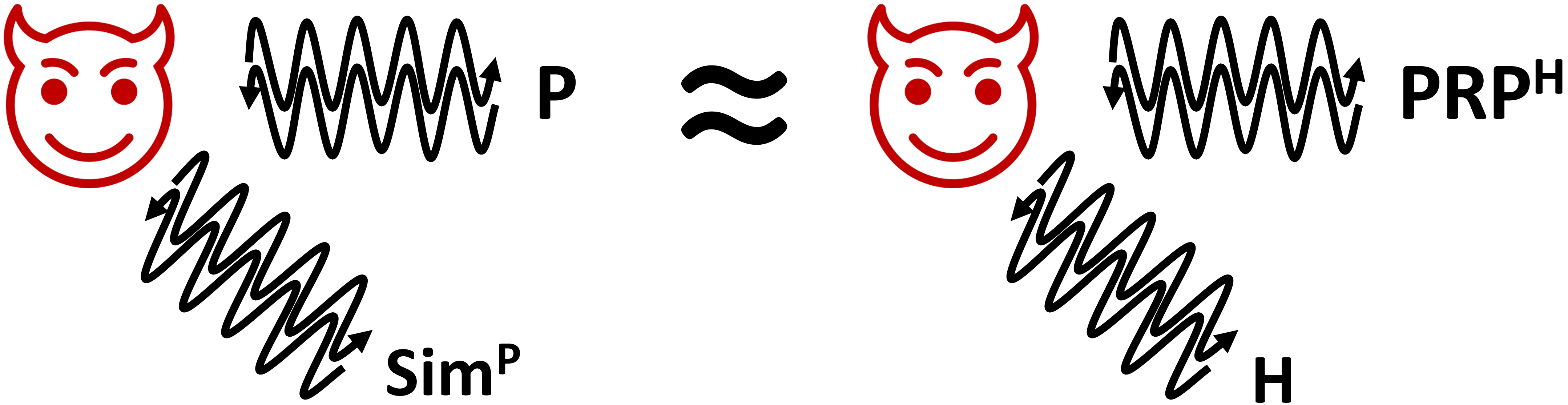
Consider a PRP built from a random function H (e.g. [Z’16]). Can this lift impossibilities?

Can clearly distinguish since no **H** on left

Attempt 2: Indifferentiability

[Maurer-Renner-Holenstein'03, Carstens-Ebrahimi-Tabia-Unruh'18, **Z**'19]

Indifferentiability sufficient for "efficient" games (e.g. most crypto)



Why compressed oracles useful?

Compressed oracles provide a *stateful* way to simulate **H**, which is often inherent for indifferentiability results

Classical world:

- Domain extension [Coron-Dodis-Malinaud-Puniya’07]
- Permutation → function [Bertoni-Daemen-Peeters-Van Assche’08]
- Function → Permutation (several-round Feistel) [Coron-Holenstein-Künzler-Patarin-Seurin-Tessaro’16]

Quantum world:

- Domain Extension [**Z**’19]
- Permutation → function [**Z**’21, Alagic-Carolan-Majenz-Tokat’15]
- **Function → Permutation completely open**

Note: for separations of complexity classes involving witnesses (e.g. NP, QMA), indifferentiability isn’t even enough, since need to simulate witness, which is inefficient

Very strong forms of indifferentiability do suffice, but in general I don’t think the “right” definition has been found