Quantum Space-Time Tradeoffs for Sponge Inversion

Joseph Carolan, Alexander Poremba, Mark Zhandry

Based on [arxiv:2403.04740] and [arxiv:2410.16595]

Motivation: Random Oracle Model

- Let $f : \{0,1\}^* \to \{0,1\}^n$ be a uniform random function
- Assume that everyone has the ability to compute f
- Assume that they can only compute f on poly(n) points
- This is called the **Random Oracle Model**
- Essentially all practical cryptosystems are analyzed in this model

Motivation: Hash Functions

- **Problem:** random oracles do not exist
- Hash functions are used as "approximate random oracles"
- Current international hash standard is SHA3
- SHA3 uses the **sponge** to achieve domain extension

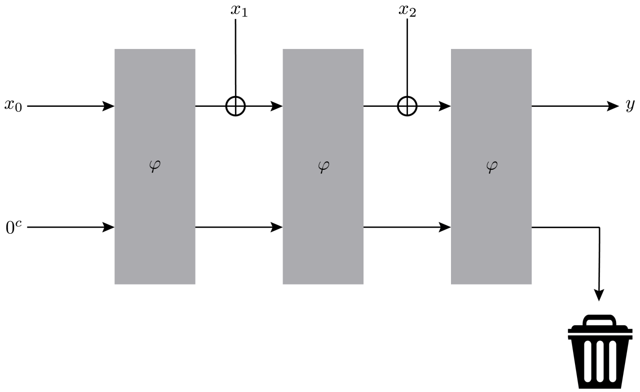


Image credit: https://www.vecteezy.com/free-vector/garbage-can, Garbage Can Vectors by Vecteezy

Sponge Construction

- Based on permutation (bijection $\varphi : \{0,1\}^n \to \{0,1\}^n$)
- Both φ and φ^{-1} have a public description
- Oracles can be implemented given this description:

$$O_\varphi |x\rangle |y\rangle = |x\rangle |y \oplus \varphi(x)\rangle$$
$$O_{\varphi^{-1}} |x\rangle |y\rangle = |x\rangle \left|y \oplus \varphi^{-1}(x)\right\rangle$$

- We model adversaries as having black-box access $O_\varphi, O_{\varphi^{-1}}$
- Treat φ as an **ideal random permutation**
- One step down the abstraction hierarchy

Sponge Security

- Classically, the sponge is “as good as” a random oracle
  ($\rightarrow$) One way, collision resistant, ...
- With **quantum** queries to φ, φ^{-1}, **nothing** is known[1]
- We have few techniques for analyzing quantum permutation problems
- Proven difficult to apply adversary/polynomial methods
- No permutation analog of compressed oracles, despite many attempts

[1] Partial progress for one-round [Z'21], [CP'24], [CPZ'24], [MMW'24] or without φ^{-1} queries [CBHSU'17], [CMSZ'19]

A Starting Point

Double Sided Zero Search (DSZS) [Unruh'21], [Unruh'23]

In: *Queries to permutation* φ *and* φ^{-1} *on* $2n$ *bits*
Out: *A "zero pair"* (x, y) *s.t.*

$$\varphi(x||0^n) = y||0^n$$

- Exhibits essential features of one-round sponge inversion
- "Even simple questions relating to (superposition access to) random permutations are to the best of our knowledge not in the scope of existing techniques, such as the following conjecture:" [Unruh'23]

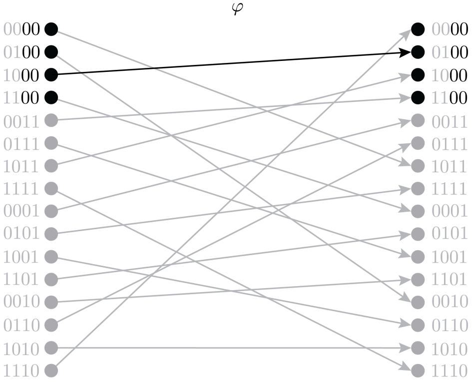
Conjecture [Unruh'21], [Unruh'23]

Solving **DSZS** requires $\Omega(\sqrt{2^n})$ quantum queries to φ, φ^{-1}

Zero Pairs Intuition

Some facts:

- Exactly one zero pair on average
- Exponentially decaying probability of more



First Result

- We prove Unruh's conjecture

Theorem [CP'24]

Finding a zero pair requires $\Omega\left(\sqrt{2^n}\right)$ *quantum queries*

Proof.

A worst-to-average case reduction:

(1) Hide zero pairs at adversarial locations

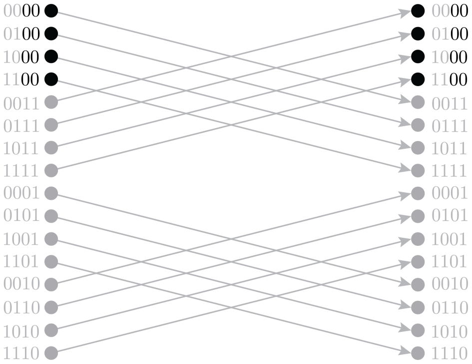
(2) Re-randomize to an average-case instance, while maintaining zero pairs (symmetrize)

□

Worst-Case Hardness

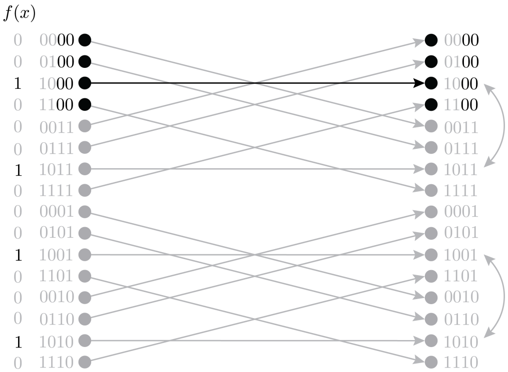
- In the worst case, solution may not exist!

$$\varphi_w(x\|y) := x\|(y \oplus 1^n)$$



Worst-Case Hardness with K solutions

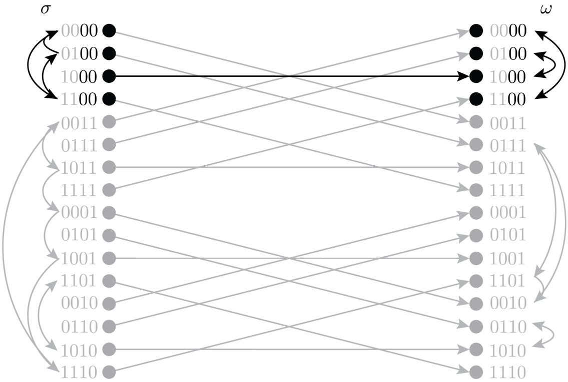
- Start from permutation with no zero pairs
- Hide zero pairs in K arbitrary positions
- Inverse queries don't help, because $\varphi_w = \varphi_w^{-1}$



Symmetrization

- Let ω, σ be random permutations that preserve suffix 0^n
- Sandwich a worst-case instance to get an average-case instance (with K zero pairs)

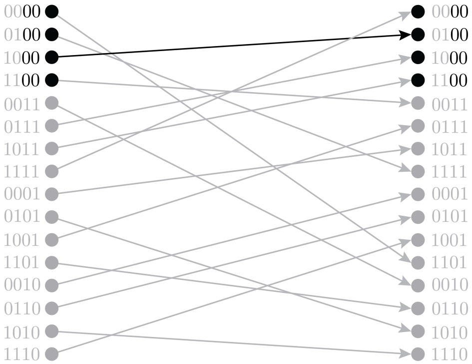
$$\varphi := \omega \circ \varphi_W \circ \sigma$$



Symmetrization

- Let ω, σ be random permutations that preserve suffix 0^n
- Sandwich a worst-case instance to get an average-case instance (with K zero pairs)

$$\varphi := \omega \circ \varphi_W \circ \sigma$$



Symmetrization Soundness

- **Main technical insight:** group theoretic picture
- Permutations preserving suffix 0^n form a subgroup
- Double cosets are permutations with fixed number of zero pairs

Symmetrization Lemma

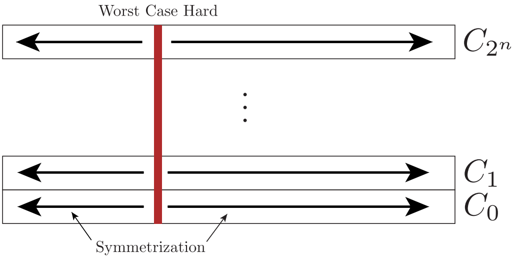
Multiplying by random elements of the left and right subgroups, re-randomizes over the double coset.

Proof Review

Theorem [CP'24]

Finding a zero pair requires $\Omega\left(\sqrt{2^n}\right)$ *quantum queries to* φ, φ^{-1}

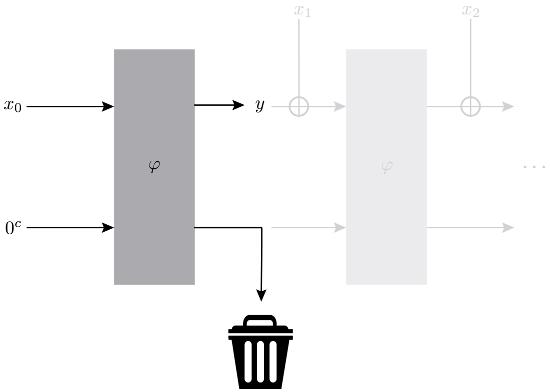
Proof(ish):



- Symmetrizing preserves the hardness!

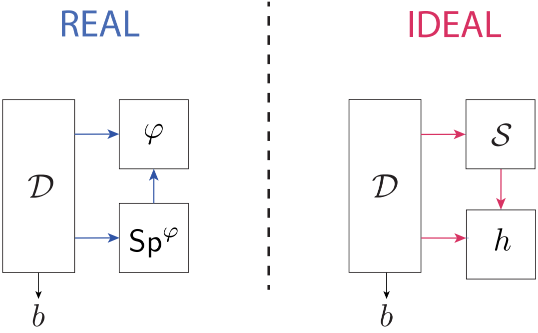
Quantum Security of the Sponge

- For simplicity, restrict to one round
- Top wire is size r = rate
- Bottom wire is size c = capacity



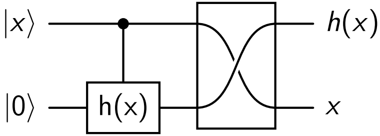
Indifferentiability Definition

- **Indifferentiability** gives a way to lift random oracle lower bounds to concrete hash functions
- Requires simulating a permutation, given just a random oracle



Sponge Indifferentiability

- We can prove indifferentiability using symmetrization
- Idea: hide a random function inside the sponge, then symmetrize
- Let us assume that $r = c$ for simplicity[2]
- To hide a function h in the Sponge:



- The sponge hash will be h

[2] we require $r \leq c$

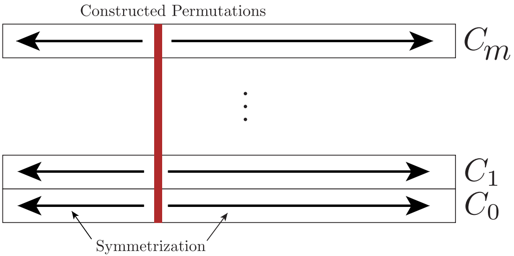
Symmetrizing the Sponge

Characterization Lemma [CPZ'24]

There exists double cosets $C_0, \dots C_m = H \backslash S_{2^n} / K$ satisfying:

$(\rightarrow)$ $\pi, \pi' \in C_j$ if and only if $\mathsf{Sp}^{\pi} = \mathsf{Sp}^{\pi'}$

- We can symmetrize φ_h while maintaining sponge



Summary of results

- Our simulator is perfectly secure
- Prior work [Zhandry'21] requires a query bound, even classically
- Our notion also captures adversaries with inefficient pre-computation
- Implies new quantum and classical results for one round Sponge:
  (1) **Tight** space-time tradeoffs for inversion
  (2) Generic, composable security in any game with **pre-computation**
  (3) **Tight** bounds for one-wayness, collision resistance, ...

Future Directions

- Indifferentiability of the full Sponge construction?
  - ($\rightarrow$) This requires overcoming the **stateful simulation** barrier
- Other applications of symmetrizing over double cosets?
- Other applications of Indifferentiability with Pre-computation?
- See also concurrent work by Majenz, Malavolta, and Walter
  - ($\rightarrow$) Similar results to [CP'24], different techniques
  - ($\rightarrow$) Talk on Friday morning!

Thank you!