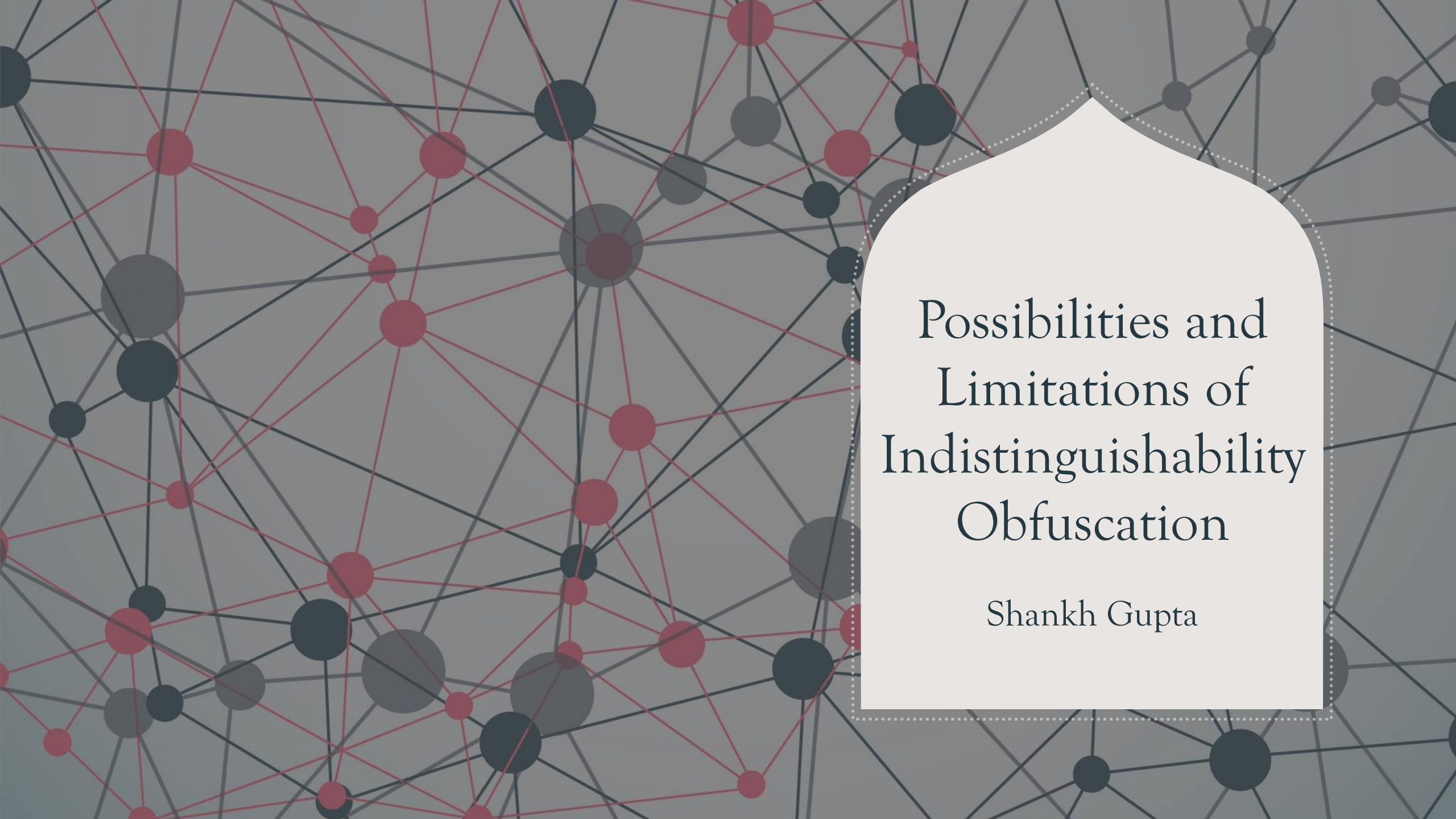
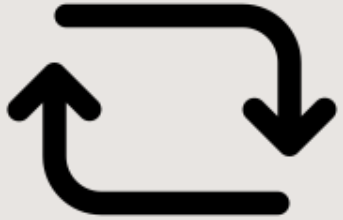


Possibilities and Limitations of Indistinguishability Obfuscation

Shankh Gupta

Contents

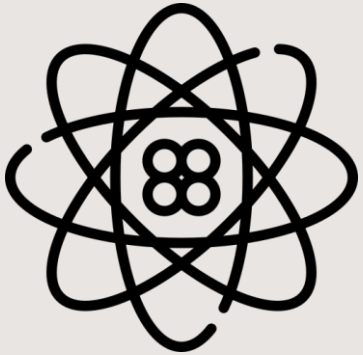


Recap:

IO, BB-Separation Result

Proof Sketch &

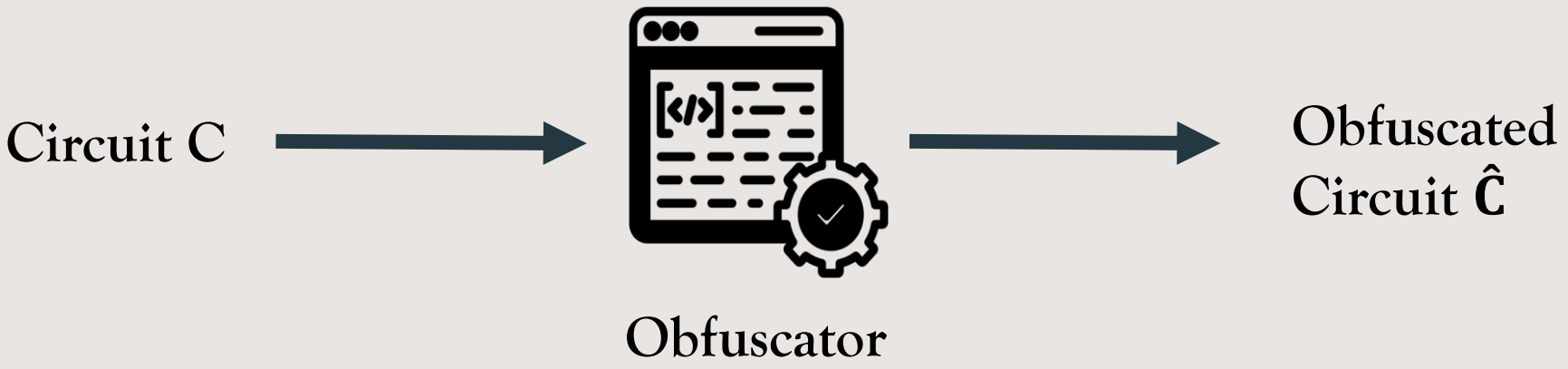
Compression Argument



Extension to Post-quantum setting.

Program Obfuscation

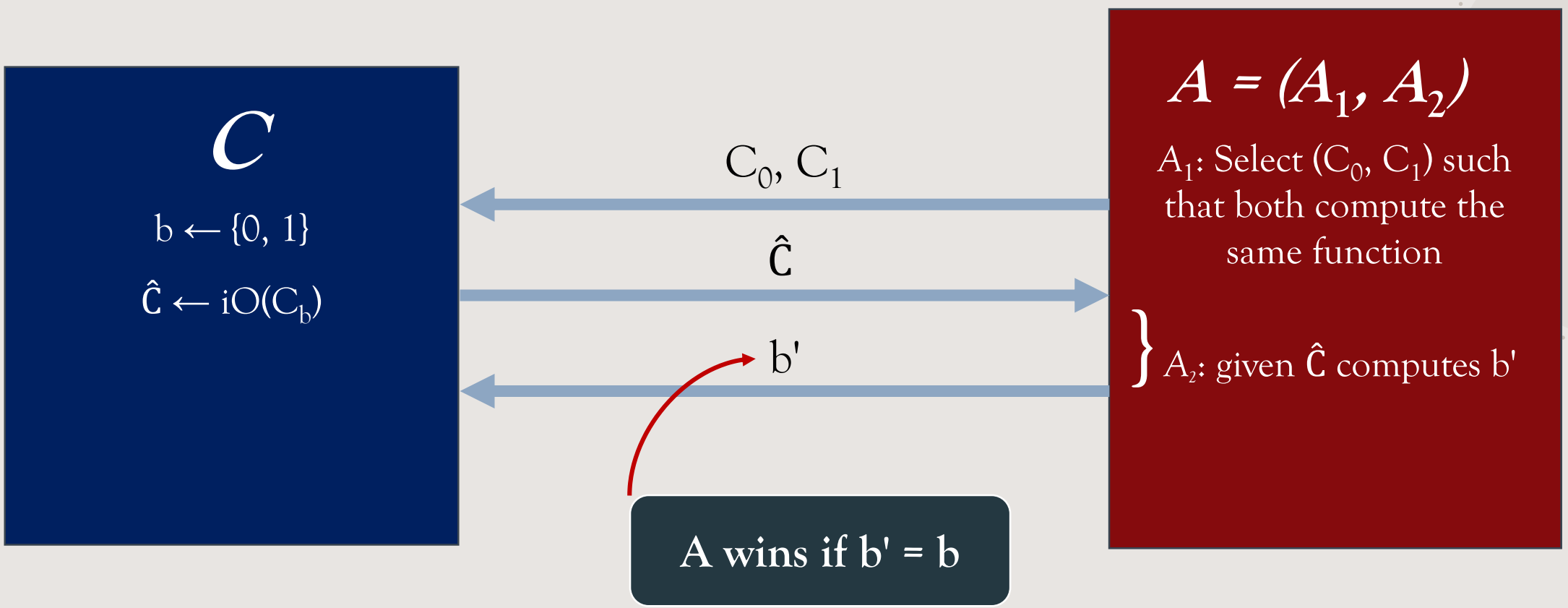
Goal : Make programs "hard to understand" while preserving functionality.



Correctness: Circuits C and $\hat{C}$ compute the same function.

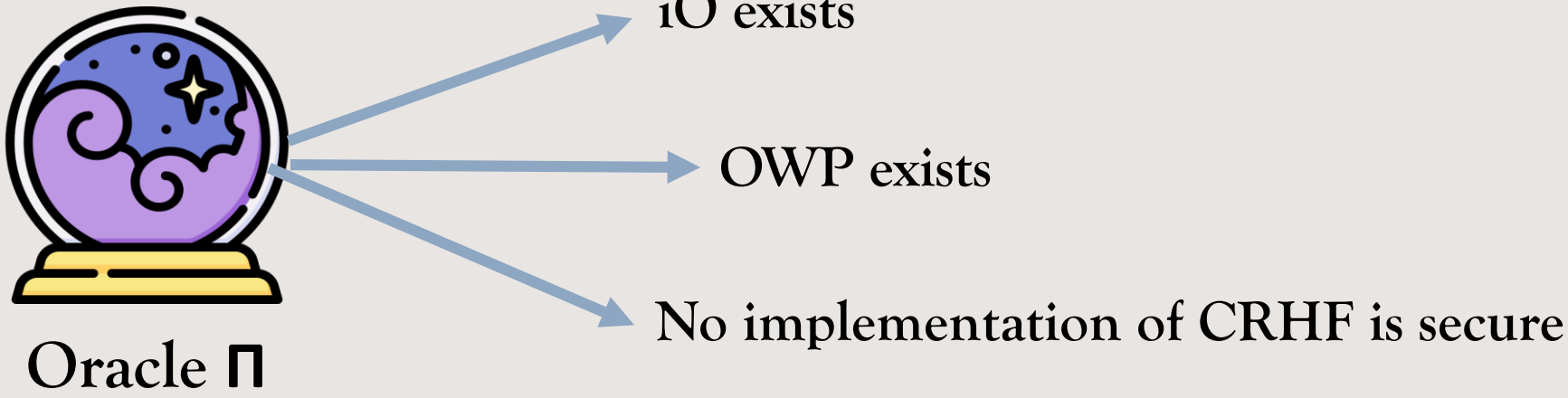
Indistinguishability Obfuscation

Security: Obfuscation of two functionally-equivalent circuits are indistinguishable.



Limitations of iO [AS15]

Theorem: There is no fully black-box construction of a CRHF from an iO for oracle-aided circuits and One-way Permutations.



Oracle Construction

Circuit C

Oracle Π

= OWP (f) + Indistinguishability Obfuscator + Collision-Finder oracle

$O(C, r)$: uniformly chosen permutation

Eval $(\hat{C}, x) \rightarrow C(x)$

⇓

Outputs a collision (w, w') w.r.t. C

Final obfuscated circuit:

Eval($\hat{C}$, .)

Proving Existence of iO

Given an obfuscation $\hat{C} \leftarrow O(C_b, r^*)$ *to the adversary:*

- **InitHit (wrt A_1):**
  - A_1 queries O on randomness r^*.
  - A_1 queries ***Eval*** or **CollFinder** which in turn makes a query to oracle O on randomness r^*.
- **r^*-hit (wrt A_2):**
  - A_2 queries O on inputs (C_0, r^*) or (C_1, r^*).
- **CollFinder-hit (wrt A_2):**
  - A_2 queries ***CollFinder*** which in turn queries O on inputs (C_0, r^*) or (C_1, r^*).

Proving Existence of iO

Claim1:
If A has some advantage in the iO game, then atleast one of initHit, r*-hit or CollFinder hit must have happened.

Claim2:
For any adversary that makes any one of the hits with some prob., there exists an adversary B that makes only r*-hit with a similar prob. (and poly blowup)

Claim3:
If an adversary makes r*-hit with significant prob., then it can be used to **compress the random permutation oracle O.**

Compressing the Oracle

•**Compression Argument:** If we can encode the truth-table of a random permutation into an encoding that can be decoded with high probability, then the size of the encoding should be almost as large as that of the truth table.

Encoding

Store a partial-truth table of O on only those inputs which A queries during it's execution.

Encoding : O restricted to all inputs except a set G. + image set of G

(select G cleverly!)

Decoding

Use A to invert image set on input values in G.

Post-quantum Setting

- The adversary can query the oracle in a superposition of inputs.
- The above claims do not directly follow in such a setting.

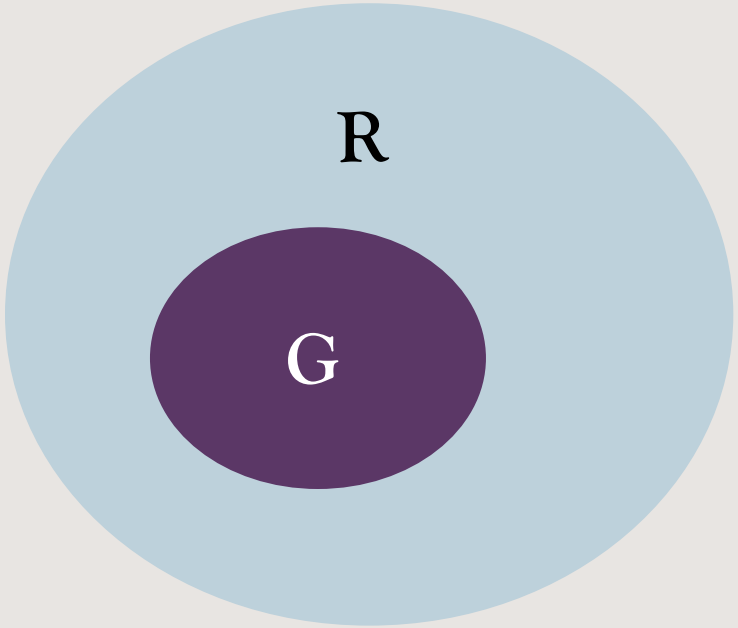
- Use the ideas of [NABT15, HY18] to extend claim 3 to post-quantum setting!

Theorem: [NABT15] Hard to invert a random permutation f with quantum-oracle access to f.

Inverting a Permutation (quantum)

Encoding:

($f : [N] \rightarrow [N]$)



G s.t. for all x in G:

- A inverts f(x) with high prob.
- Query magnitude of A on any element in R\x is sufficiently small

Output: $f|_{\text{restricted to } [N]\setminus G} + f(G)$

Decoding:

- Use A to invert values in f(G) relative to some f' that agrees with A on values in [N]\G.
- A still inverts f(G) because the query-magnitude on input R\x is small. (Swapping-Lemma).

Future work

- Prove Claim 3 in the post-quantum setting using ideas from NABT15, HY18.
- Try to extend Claims 1 and 2 in the quantum-oracle setting as well.

Thank You!