Building Hard Problems by Combining Easy Ones, Revisited

Yael Eisenberg[1], Christopher Havens[2], Alexis Korb[3], Elio Merolle[4] and Amit Sahai[5]

[1] Cornell University, Ithaca, USA
[2] Department of Computer Science, UCLA; Washington Corrections Center; and Prison Mathematics Project, USA
[3] UCLA, Los Angeles, USA
[4] UCLA, Los Angeles, USA
[5] UCLA, Los Angeles, USA

Abstract. We continue the study initiated by Ghosal and Sahai (ISIT 23) of generating hard functions by composing simpler functions that are both easy to evaluate and to invert, modeled as random functions and their inverses.
We establish the following theorem: Let $\mathsf{O}_0, \mathsf{O}_1, \mathsf{R}$ be random functions from $\{0,1\}^n$ to $\{0,1\}^m$, $m, n \in \mathbb{N}$ and $m \geq n \geq 10$. For all distinguishers D making at most q oracle queries, where $q \leq \dfrac{2^n}{n+1}$, there exists a $\mathsf{poly}(n, m, q)$-time oracle simulator $\mathsf{Sim}^{(\cdot)}$ such that:

$$\left|\Pr\left[\mathsf{D}^{(\mathsf{O}_0+\mathsf{O}_1),(\mathsf{O}_0,\mathsf{O}_1,\mathsf{O}_0^{-1},\mathsf{O}_1^{-1})}(1^n)=1\right]-\Pr\left[\mathsf{D}^{\mathsf{R},\mathsf{Sim}^{\mathsf{R}}}(1^n)=1\right]\right| \leq \frac{8nq^2}{2^m}+\frac{7n^2q^2}{2^n}.$$

In contrast to prior work, we consider and prove indifferentiability in all cases where $m \geq n$ and n is sufficiently large.

Keywords: Indifferentiability · Random Oracles · Composition of Functions

E-mail: ye45@cornell.edu (Yael Eisenberg), chavens28@gmail.com (Christopher Havens), alexiskorb@cs.ucla.edu (Alexis Korb), eliom@ucla.edu (Elio Merolle), sahai@cs.ucla.edu (Amit Sahai)

1 Introduction

Cryptography is built upon the conjectured hardness of various mathematical problems which are necessary for constructing cryptographic primitives such as public-key encryption [GM19], secure two-party computation [Yao82], and pseudo-random generators [BM84]. Although we have known that there are an abundance of hard problems since at least the work of Claude Shannon [Sha49], formally proving the hardness of even one concrete problem has proven elusive. Indeed, one of the most foundational hardness-based objects is a *one-way function* f (see e.g. [Gol01]), which satisfies:

1) Given any valid input x, it is easy to compute $f(x)$

2) Given $f(x)$ for a randomly chosen x in the domain of f, it is hard to compute any x' such that $f(x') = f(x)$.

However, showing that any explicit one-way function exists would imply that **NP** is not contained in **BPP**, and thus we have little present hope of explicitly proving that one-way functions exist. Easy functions, on the other hand, satisfy the following properties:

1) Given any valid input x, it is easy to compute $f(x)$.

2) Given $f(x)$ for a randomly chosen x in the domain of f, it is easy to compute some x' such that $f(x') = f(x)$.

In other words, these functions are easy to evaluate and allow efficient computation of a preimage for any given output.

A simple example of an "easy" function is $f : \{0,1\}^4 \to \{0,1\}^2$ defined by XORing consecutive pairs of bits:

$$f(x_1x_2x_3x_4) = (x_1 \oplus x_2,\ x_3 \oplus x_4).$$

Clearly, f can be evaluated efficiently. To find a preimage of an arbitrary $y = y_1y_2$, one can, for instance, set $x_2 = x_4 = 0$ and $x_1 = y_1$, $x_3 = y_2$, producing $x' = x_1x_2x_3x_4$ such that $f(x') = y$. This demonstrates the "easy" invertibility of the function.

The Ghosal-Sahai Theoretical Model. Confronted with this barrier, Ghosal and Sahai initiated an investigation of "recipes" for constructing secure one-way functions by combining two insecure and unrelated "easy" functions [GS23]. This study was well-motivated by the desire to understand principles that underlie existing candidate hard problems (see [GS23] for further details on motivation). More specifically, Ghosal and Sahai put forward a model where each "easy" function is modeled as an easy-to-invert random function, given by a random oracle $f : \{0,1\}^n \to \{0,1\}^m$ and its inverse oracle $f^{-1} : \{0,1\}^m \to \mathcal{P}\big(\{0,1\}^n\big)$, where for any set A we define $\mathcal{P}(A)$ to be the power set of A. Thus, for example, setting $n = m = 3$ and fixing a particular random choice for the oracle f, if in this model an adversary queries $f(000)$, it could receive the output value 010, but if it then queries $f^{-1}(010)$, it might receive the output $\{000, 011\}$, because 010 has two preimages.

Their work then considered two such "easy" functions $f_1 : \{0,1\}^n \to \{0,1\}^m$ and $f_2 : \{0,1\}^n \to \{0,1\}^m$, when m is sufficiently larger than n. They then showed that the function $f(x) = f_1(x) + f_2(x)$ is provably one-way, even given oracle access to f_1, f_2 and their respective inverses. Specifically, this new function f is *indifferentiable* from a random oracle, as captured by the theorem below:

Theorem [Imported from [GS23]] *Let* $\mathsf{O}_0, \mathsf{O}_1, \mathsf{R}$ *be random functions from* $\{0,1\}^n$ *to* $\{0,1\}^m$*, where* $m = n + t(n)$, $m, n, t \in \mathbb{N}$. *For all polynomial-query-bounded distinguishers*

D *making at most* $q = \mathsf{poly}(n)$ *oracle queries, there exists a poly-time oracle simulator* $\mathsf{Sim}^{(\cdot)}$ *and a constant* $c > 0$ *such that*

$$\left|\Pr\left[\mathsf{D}^{(\mathsf{O}_0+\mathsf{O}_1),(\mathsf{O}_0,\mathsf{O}_1,\mathsf{O}_0^{-1},\mathsf{O}_1^{-1})}(1^n)=1\right]-\Pr\left[\mathsf{D}^{\mathsf{R},\mathsf{Sim}^{\mathsf{R}}}(1^n)=1\right]\right| \leq \frac{cq}{2^t}+\frac{6q^2}{2^{n+t}}.$$

Note that D*'s advantage above becomes negligible only when* $t = \omega(\log n)$*.*

Unfortunately, their result only succeeds when the functions f_1 and f_2 are sufficiently expanding. Specifically, they require that with high probability, a random value in the codomain of f_1 and f_2 does not have a preimage, and they use this fact to handle *all* nontrivial calls to inverse oracles by the adversary in their proof of indifferentiability.

In our work, we continue the study of the Ghosal-Sahai Model for the same recipe, and we ask if the main limitation of their work is a fundamental barrier. Indeed, since it is known that a length-preserving random oracle by itself is a one-way function, we might also expect similar results to hold for any $m \geq n$. Attacking this problem not only addresses an open problem of [GS23], it also furthers our understanding of the fundamental question of what input/output sizes are necessary for building one-way functions out of simpler easy-to-invert functions.

In this paper, we solve the open problem in the affirmative and establish a similar theorem to the theorem of [GS23] stated above, which holds even when $t = 0$, that is, whenever $f_1, f_2 : \{0,1\}^n \to \{0,1\}^m$ are random functions satisfying $m \geq n$. We prove the following:

Theorem 1. *Let* $\mathsf{O}_0, \mathsf{O}_1, \mathsf{R}$ *be random functions from* $\{0,1\}^n$ *to* $\{0,1\}^m$*,* $m, n \in \mathbb{N}$ *and* $m \geq n \geq 10$*. For all distinguishers* D *making at most* q *oracle queries, where* $q \leq \dfrac{2^n}{n+1}$*, there exists a* $\mathsf{poly}(n, m, q)$*-time oracle simulator* $\mathsf{Sim}^{(\cdot)}$ *such that:*

$$\left|\Pr\left[\mathsf{D}^{(\mathsf{O}_0+\mathsf{O}_1),(\mathsf{O}_0,\mathsf{O}_1,\mathsf{O}_0^{-1},\mathsf{O}_1^{-1})}(1^n)=1\right]-\Pr\left[\mathsf{D}^{\mathsf{R},\mathsf{Sim}^{\mathsf{R}}}(1^n)=1\right]\right| \leq \frac{8nq^2}{2^m}+\frac{7n^2q^2}{2^n}.$$

Our primary technical contribution is a technique for simulating queries to the inverse oracles of f_1 and f_2. In the prior work, since the functions were sufficiently expanding, almost all inverse queries could be answered with $\perp$, since most elements of the codomain would not have a preimage. In contrast, if $m - n$ is small, then we must keep careful track of inverse queries to ensure consistency in the simulation.

Note that our setting allows there to be a non-negligible probability that a queried value $b \in \{0,1\}^m$ has multiple preimages, although the probability that any such queried value b has more than n preimages is still negligible. In the previous work of [GS23], in contrast, the probability that any queried value b had more than one preimage was negligible. In our simulation, when the adversary asks for the inverse of b, we actually provide a simulation that provides *all* preimages of b to the adversary, not just one. This is an additional technical contribution of our work.

We leave open the problem of developing a new analysis for the setting of *shrinking* functions, where $m < n$, and investigate whether the sum (or composition) of such functions remains computationally indistinguishable from a truly random function. This would complete the picture by covering both expanding and shrinking regimes.

Related Work. In other related work, [GBJ+23, Gun22, BN18, MP15, MPN10] consider indifferentiability of sums of *permutations*. In contrast, our functions are not permutations. In particular, in our setting, even if the domain and range of our random functions are the same size, with high probability it will be true that a constant fraction of elements in the range have zero preimages, and a constant fraction have more than one preimage. As previously mentioned, this alters the simulation strategy that one needs to construct. We

also note that circuits that behave like random permutations are quite rare and can be difficult to construct; in contrast functions that are modeled as random oracles are much more common. This provides further motivation for our line of study.

2 Preliminaries

Definition 1 (Polynomial-Query-Bounded Oracle Turing Machine)**.** We say that an oracle Turing machine $T^{(\cdot)}$ is *polynomial-query-bounded* if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that for any input $x \in \{0,1\}^*$ and for any oracle O, the execution of $T^O(x)$ makes at most $p(|x|)$ many queries to O. ◊

Definition 2 (Indifferentiability [MRH04, DKT16])**.**] Let $\mathbf{C} : \{0,1\}^n \to \{0,1\}^{m(n)}$ be a construction having access to an ideal primitive $\mathsf{F} : \{0,1\}^{p(n)} \to \{0,1\}^{q(n)}$ and implements a functionality based on F, where $p(n), q(n), m(n) = poly(n)$. We say that $\mathbf{C}$ is *indifferentiable* from a random function $RO : \{0,1\}^n \to \{0,1\}^m$, if there is a poly-time simulator Sim with oracle access to RO such that for all polynomial-query-bounded distinguishers D, we have

$$\left| \Pr\left[D^{\mathbf{C}^F,F}(1^n)\right] - \Pr\left[D^{RO,\mathsf{Sim}^{RO}}(1^n)\right] \right| \tag{1}$$

is negligible. ◊

3 Our Result

In this section, we prove our main result:

Theorem 1. (restated) *Let* $\mathsf{O}_0, \mathsf{O}_1, \mathsf{R}$ *be random functions from* $\{0,1\}^n$ *to* $\{0,1\}^m$*,* $m, n \in \mathbb{N}$ *and* $m \geq n \geq 10$*. For all distinguishers* D *making at most* q *oracle queries, where* $q \leq \dfrac{2^n}{n+1}$*, there exists a* $\mathsf{poly}(n,m,q)$*-time oracle simulator* $\mathsf{Sim}^{(\cdot)}$ *such that:*

$$\left|\Pr\left[\mathsf{D}^{(\mathsf{O}_0+\mathsf{O}_1),(\mathsf{O}_0,\mathsf{O}_1,\mathsf{O}_0^{-1},\mathsf{O}_1^{-1})}(1^n)=1\right] - \Pr\left[\mathsf{D}^{\mathsf{R},\mathsf{Sim}^{\mathsf{R}}}(1^n)=1\right]\right| \leq \frac{8nq^2}{2^m} + \frac{7n^2q^2}{2^n}.$$

Our theorem makes use of the following oracles:

- $\mathsf{O}_0, \mathsf{O}_1, \mathsf{R}$: random oracles from $\{0,1\}^n$ to $\{0,1\}^m$.
- $\mathsf{O}_0^{-1}, \mathsf{O}_1^{-1}$: the inverse oracles for O_0 and O_1 respectively. Here, $\mathsf{O}_i^{-1}(b) = S$ means that S is the (possibly empty) set of *all* preimages of b under O_i.

These oracles are then grouped into a left oracle LOra and right oracle ROra as follows:

- LOra: takes as input a value a and returns an output b.
  - In the real world, $\mathsf{LOra}(a) = \mathsf{O}_0(a) + \mathsf{O}_1(a)$.
  - In the ideal world, $\mathsf{LOra}(a) = R(a)$.
- ROra: takes as input an oracle name $\mathsf{OraName} \in \{\mathsf{O}_0, \mathsf{O}_1, \mathsf{O}_0^{-1}, \mathsf{O}_1^{-1}\}$ and an input x and returns a value y.
  - In the real world, $\mathsf{ROra}(\mathsf{OraName}, x)$ returns the output of running the specified oracle on x.
  - In the ideal world, $\mathsf{ROra}(\mathsf{OraName}, x)$ will be answered by a simulator Sim^R which has oracle access to R.

In our security proof, we will use the following registers and notation for keeping track of oracle queries.

- Reg_i will contain pairs (a, b_i) corresponding to queries to O_i. We will add (a, b_i) to Reg_i if we have determined that $\mathsf{ROra}(\mathsf{O}_i, a) = b$.
  - $\mathsf{Dom}(\mathsf{Reg}_i) := \{a \mid \exists b \text{ s.t. } \mathsf{Reg}_i(a) = b\}$. These are the set of domain values of O_i whose image has been determined.
  - $\alpha_i := |\mathsf{Dom}(\mathsf{Reg}_i)|$.
- InvReg_i will contain pairs (S, b) corresponding to queries to O_i^{-1}. We will add (S, b) to InvReg_i if we have determined that $\mathsf{ROra}(\mathsf{O}_i^{-1}, b) = S$.
  - $\mathsf{Im}(\mathsf{InvReg}_i) := \{b \mid \exists S \text{ s.t. } \mathsf{InvReg}_i(S) = b\}$. These are the set of codomain values of O_i whose preimage has been completely determined.
  - $\beta_i := |\mathsf{Im}(\mathsf{InvReg}_i)|$.
- LReg will contain values a corresponding to queries to the left oracle LOra.
- FReg will contain values a corresponding to inputs to O_0 or O_1 that we have ensured are consistent with the output of the left oracle LOra.

Whenever we update InvReg_i, we will also immediately update Reg_i to be consistent with InvReg_i. On the other hand, since adding (a, b_i) to Reg_i does not necessarily mean that a is the *only* preimage of b_i, we will not immediately update InvReg_i on updates to Reg_i. (This will also be necessary later on for efficient simulation.) We will only update InvReg_i on queries to O_i^{-1}, at which point we will take into account any prior relevant updates to Reg_i.

Proof of Theorem 1. We present the proof using a hybrid argument. For notational simplicity, we use $(\mathsf{LOra}_i, \mathsf{ROra}_i)$ to denote the two oracles accessed by the distinguisher D in Hybrid i. Our first hybrid H_1 corresponds to the real world experiment whereas our last hybrid H_9 is the ideal world experiment. The simulator is described in H_9. We will have intermediate results that demonstrate indistinguishability between H_i and H_{i+1} ultimately allowing us to conclude that H_1 is indistinguishable from H_9 via triangle inequality.

Proof Overview. Our first hybrid H_1 is simply the real-world experiment described above. Through hybrids H_2 and H_3 the only thing we change are: (1) We introduce the book-keeping registers described above to keep track of adversary oracle queries; and (2) We also introduce a new process we call FReact, to ensure that whenever the adversary calls O_i, where $i \in \{0, 1\}$, on some input $a \in \{0, 1\}^n$, we actually also pretend that the adversary queried $\mathsf{O}_{1-i}(a)$ as well. This will be crucial later when we are trying to simulate these oracles O_0 and O_1 using only a random oracle R such that $R(a)$ always equals simulated $\mathsf{O}_0(a) + \mathsf{O}_1(a)$. We then proceed to hybrid H_4, which is still a perfect emulation of the real world, except that it utilizes a change in perspective that we call *lazy sampling*. By lazy sampling, we mean that instead of sampling the entire random oracles O_0 and O_1 at the beginning of the hybrid, we instead "populate" the oracles as necessary as new adversary queries come in, thus beginning our process of simulating the oracles O_0 and O_1. Unfortunately, the cost doing this lazy sampling perfectly is efficiency, since it takes $O(2^n)$ time to determine exactly how many preimages any value $b \in \{0, 1\}^m$ would have with respect to a random oracle mapping $\{0, 1\}^n$ to $\{0, 1\}^m$. We fix this efficiency problem in hybrid H_5, using the observation that it is exponentially unlikely that any queried $b \in \{0, 1\}^m$ would have more than n preimages. Having set the stage in this way, in hybrid H_6 we first start making use of the external random oracle R: namely, we change the way FReact acts so that if the adversary queried $\mathsf{O}_i(a)$, we assign the value $\mathsf{O}_{1-i}(a) = R(a) - \mathsf{O}_i(a)$, ensuring that $R(a)$ always equals simulated $\mathsf{O}_0(a) + \mathsf{O}_1(a)$. Finally,

in our remaining series of hybrids H_7 through H_9, we "undo" some of the book-keeping needed to yield a full simulator.

We note that our proof temporarily departs from efficiency at H_3. This does not compromise the overall argument: the inefficiency is introduced purely as an analytical device to simplify reasoning about oracle behavior. Conceptually, one can regard the inefficient hybrids as defining idealized distributions that we later approximate by efficient constructions. Since indistinguishability is only required to hold between pairs of efficient hybrids (e.g., between H_2 and H_5), and both endpoints of our hybrid chain are efficient, the final conclusion about PPT indistinguishability remains sound.

- **Hybrid H_1**
  This is the real world experiment where D interacts with oracles $(\mathsf{LOra}_1, \mathsf{ROra}_1) = ((\mathsf{O}_0 + \mathsf{O}_1), (\mathsf{O}_0, \mathsf{O}_0^{-1}, \mathsf{O}_1, \mathsf{O}_1^{-1}))$. This hybrid, and the next two hybrids, all begin by first randomly sampling the oracles O_0 and O_1.
  - $\mathsf{LOra}_1(a)$: Return $\mathsf{O}_0(a) + \mathsf{O}_1(a)$ where O_0 and O_1 are the real world random oracles.
  - ROra_1: Answer queries of the form $\mathsf{O}_0(a)$, $\mathsf{O}_0^{-1}(b)$, $\mathsf{O}_1(a)$, $\mathsf{O}_1^{-1}(b)$ according to the corresponding real world random oracles O_0, O_1, O_0^{-1}, O_1^{-1}.
- **Hybrid H_2**
  We now answer queries to the left oracle LOra by utilizing the right oracle ROra.
  - $\mathsf{LOra}_2(a)$: Return $\mathsf{ROra}_2(\mathsf{O}_0, a) + \mathsf{ROra}_2(\mathsf{O}_1, a)$.
  - ROra_2: Answer queries of the form $\mathsf{O}_0(a)$, $\mathsf{O}_0^{-1}(b)$, $\mathsf{O}_1(a)$, $\mathsf{O}_1^{-1}(b)$ according to the corresponding real world random oracles O_0, O_1, O_0^{-1}, O_1^{-1}.

  Lemma 1. *For all $n \in \mathbb{N}$ and all oracle Turing Machines D,*

  $$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_1,\mathsf{ROra}_1}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_2,\mathsf{ROra}_2}(1^n) = 1]\right| = 0$$

  Proof. The hybrids have identical behavior, so the distinguishing advantage is 0. □
- **Hybrid H_3**
  This is the same as the previous hybrid except that
  - we respond to queries to O_i^{-1} by *inefficiently* querying O_i to find all preimages of the specified value,
  - we record queries to oracles O_i and O_i^{-1} in registers Reg_i and InvReg_i respectively, and respond to previously queried values with the stored register values,
  - whenever we record a query to O_i, we also issue the corresponding query to O_{1-i} (if it has not already been made) and register the shared input in FReg. This is achieved by running FReact_3.

  This formally sets up the paired queries so that $\mathsf{LOra}(a) = \mathsf{O}_0(a) + \mathsf{O}_1(a)$ is well-defined. Conceptually, by defining LOra as the sum of the two component oracles, we effectively achieve the same effect as FReact_3 without actually running it: consistency is guaranteed by construction, so there is no risk of inconsistencies at this stage. We include FReact_3 here primarily for clarity and to prepare for later hybrids, starting with H_7, when $\mathsf{LOra}(a)$ will be defined via an independent function $R(a)$ and explicit corrective action becomes necessary. We remark that while these changes do not affect the output of the oracles, our hybrid is now *inefficient*.
  - $\mathsf{LOra}_3(a)$: Return $\mathsf{ROra}_3(\mathsf{O}_0, a) + \mathsf{ROra}_3(\mathsf{O}_1, a)$.

– ROra_3:
  * On queries of the form $\mathsf{O}_i(a)$ for some $i \in \{0,1\}$:
    1. If $(a, b_i) \in \mathsf{Reg}_i$ for some b_i, return b_i.
    2. Else:
       (a) Set $b_i = \mathsf{O}_i(a)$.
       (b) $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b_i)\}$.
       (c) Run $\mathsf{FReact}_3(i, (a, b_i))$.
       (d) Return b_i.
  * On queries of the form $\mathsf{O}_i^{-1}(b)$ for some $i \in \{0,1\}$:
    1. If $(S, b) \in \mathsf{InvReg}_i$ for some S, return S.
    2. Else:
       (a) Compute the set S of all $a \in \{0,1\}^n$ such that $\mathsf{O}_i(a) = b$.
       (b) $\mathsf{InvReg}_i = \mathsf{InvReg}_i \cup \{(S, b)\}$.
       (c) For $a \in S$,
          i. $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b)\}$.
          ii. Run $\mathsf{FReact}_3(i, (a, b))$.
       (d) Return S.

where we define:

$\mathsf{FReact}_3(i, (a, b_i))$:
1. If $a \notin \mathsf{FReg}$:
   (a) Set $b_{1-i} = \mathsf{O}_{1-i}(a)$.
   (b) $\mathsf{Reg}_{1-i} = \mathsf{Reg}_{1-i} \cup \{(a, b_{1-i})\}$.
   (c) $\mathsf{FReg} = \mathsf{FReg} \cup \{a\}$.

Lemma 2. *For all $n \in \mathbb{N}$ and all oracle Turing Machines D,*

$$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_2, \mathsf{ROra}_2}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_3, \mathsf{ROra}_3}(1^n) = 1]\right| = 0$$

Proof. The hybrids have identical behavior, so the distinguishing advantage is zero. □

- **Hybrid H_4**
  In this hybrid, we answer oracle queries via lazy sampling. From this hybrid and onwards, we *do not* begin the hybrid by sampling the random oracles O_0 and O_1. We respond to new forward queries to O_0 and O_1 by randomly sampling an output value from the set of outputs whose preimage set has not yet been fully determined. For new queries to O_0^{-1} and O_1^{-1}, we first determine how large a set of preimages we should return (in addition to those known so far), and then randomly sample these preimages from the set of unused domain values.

  – $\mathsf{LOra}_4(a)$: Return $\mathsf{ROra}_4(\mathsf{O}_0, a) + \mathsf{ROra}_4(\mathsf{O}_1, a)$.
  – ROra_4:
    * On queries of the form $\mathsf{O}_i(a)$ for some $i \in \{0,1\}$:
      1. If $(a, b_i) \in \mathsf{Reg}_i$ for some b_i, return b_i.
      2. Else:
         (a) $b_i \xleftarrow{\$} \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_i)$.
         (b) $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b_i)\}$.

(c) Run $\mathsf{FReact}_4(i,(a,b_i))$.
(d) Return b_i.

∗ On queries of the form $\mathsf{O}_i^{-1}(b)$ for some $i \in \{0,1\}$:
1. If $(S,b) \in \mathsf{InvReg}_i$ for some S, return S.
2. Else:
(a) Let $S = \{x_1, x_2, \ldots, x_d\}$ be the set (possibly empty) of all elements x such that $\mathsf{Reg}_i(x) = b$.
(b) Choose k from $\mathrm{Bin}(2^n - \alpha_i, (2^m - \beta_i)^{-1})$ where $\alpha_i = |\mathsf{Dom}(\mathsf{Reg}_i)|$ and $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$.
(c) If $k > 0$,
i. Choose k domain elements $a_1, \ldots, a_k \in \{0,1\}^n \setminus \mathsf{Dom}(\mathsf{Reg}_i)$ uniformly at random, without replacement.
ii. $S = S \cup \{a_1, \ldots, a_k\}$.
(d) $\mathsf{InvReg}_i = \mathsf{InvReg}_i \cup \{(S,b)\}$.
(e) For $a \in S$,
i. $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a,b)\}$.
ii. Run $\mathsf{FReact}_4(i,(a,b))$.
(f) Output S.

where we define

$\mathsf{FReact}_4(i,(a,b_i))$:

1. If $a \notin \mathsf{FReg}$:
(a) $b_{1-i} \xleftarrow{\$} \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_{1-i})$.
(b) $\mathsf{Reg}_{1-i} = \mathsf{Reg}_{1-i} \cup \{(a, b_{1-i})\}$.
(c) $\mathsf{FReg} = \mathsf{FReg} \cup \{a\}$

Lemma 3. *For all $n \in \mathbb{N}$ and all oracle Turing Machines D,*

$$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_3,\mathsf{ROra}_3}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_4,\mathsf{ROra}_4}(1^n) = 1]\right| = 0$$

We will argue that hybrid H_4 is a perfect simulation of the previous hybrid H_3. To do so, we will first prove the following lemmas:

Lemma 4. *After any sequence of queries in H_3, the number of random oracles $\mathsf{O}_i : \{0,1\}^n \to \{0,1\}^m$ consistent with Reg_i and InvReg_i is $(2^m - \beta_i)^{2^n - \alpha_i}$, where $\alpha_i = |\mathsf{Dom}(\mathsf{Reg}_i)|$ and $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$.*

Proof. There are 2^m codomain elements of which β_i have complete preimages; therefore there are $2^m - \beta_i$ remaining codomain elements. Similarly, there are $2^n - \alpha_i$ remaining domain elements whose image has not been determined. Therefore, since no assumptions about bijectivity are made (we only require the functions to be random), each of the remaining $2^n - \alpha_i$ elements admits $2^m - \beta_i$ possible choices, yielding $(2^m - \beta_i)^{2^n - \alpha_i}$ distinct consistent oracles. □

Lemma 5. *Suppose that in H_3, no previous query to O_i or O_{1-i} on x has been made. On a query $\mathsf{O}_i(x)$ or $\mathsf{O}_{1-i}(x)$, the output will be one of the values $y_i \in \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_i)$ or $y_{1-i} \in \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_{1-i})$. For any such (y_i, y_{1-i}) pair, the probability that $\mathsf{O}_i(x) = y_i$ and $\mathsf{O}_{1-i}(x) = y_{1-i}$ is equal to $\dfrac{1}{(2^m - \beta_i)(2^m - \beta_{1-i})}$, where $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$ and $\beta_{1-i} = |\mathsf{Im}(\mathsf{InvReg}_{1-i})|$*

Proof. For O_i, by Lemma 4, there are $(2^m - \beta_i)^{2^n - \alpha_i}$ remaining oracles that are consistent with the queries already made. Now suppose $\mathsf{O}_i(x) = y_i$ for some fixed value $y_i \in \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_i)$. Of the $(2^m - \beta_i)^{2^n - \alpha_i}$ possible oracles, there are still $(2^m - \beta_i)^{2^n - \alpha_i - 1}$ remaining oracles after choosing y_i. Thus, the probability that $\mathsf{O}_i(x) = y_i$ is

$$\frac{(2^m - \beta_i)^{2^n - \alpha_i - 1}}{(2^m - \beta_i)^{2^n - \alpha_i}} = \frac{1}{2^m - \beta_i}.$$

Similarly, for O_{1-i}, by Lemma 4, there are a total of $(2^m - \beta_{1-i})^{2^n - \alpha_{1-i}}$ possible oracles consistent with the queries made so far. Thus, the probability that $\mathsf{O}_{1-i}(x) = y_{1-i}$ for some fixed value $y_{1-i} \in \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_{1-i})$is

$$\frac{(2^m - \beta_{1-i})^{2^n - \alpha_{1-i} - 1}}{(2^m - \beta_{1-i})^{2^n - \alpha_{1-i}}} = \frac{1}{2^m - \beta_{1-i}}.$$

Therefore, we have

$$\begin{aligned} Pr[\mathsf{O}_i(x) = y_i] \text{ and } Pr[\mathsf{O}_{1-i}(x) = y_{1-i}] &= \frac{1}{2^m - \beta_i} \times \frac{1}{2^m - \beta_{1-i}} \\ &= \frac{1}{(2^m - \beta_i)(2^m - \beta_{1-i})} \end{aligned}$$

□

Lemma 6. *Let $\mathsf{O}_i^{-1}(y)$ be a backwards query in H_3 where no such query has yet been made. Suppose that there exists a set $\{x_1, x_2, \ldots x_d\}$ such that $\mathsf{Reg}_i(x_t) = y$ for $1 \le t \le d$.*

The conditional probability that y has $\ell + d$ preimages given what has already been queried is

$$\binom{2^n - \alpha_i}{\ell} \left(\frac{1}{2^m - \beta_i}\right)^{\ell} \left(1 - \frac{1}{2^m - \beta_i}\right)^{2^n - \alpha_i - \ell}.$$

where $\alpha_i = |\mathsf{Dom}(\mathsf{Reg}_i)|$ and $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$.

Proof. Observe that there are $\binom{2^n - \alpha_i}{\ell}$ subsets of exactly ℓ elements taken from a domain of size $2^n - \alpha_i$. There are now $2^n - \alpha_i - \ell$ remaining domain values which can be mapped to the remaining $2^m - \beta_i - 1$ codomain values. This can be done in $(2^m - \beta_i - 1)^{2^n - \alpha_i - \ell}$ ways, making

$$\binom{2^n - \alpha_i}{\ell} (2^m - \beta_i - 1)^{2^n - \alpha_i - \ell}$$

oracles for which there are exactly ℓ additional preimages for $\mathsf{O}_i^{-1}(y)$.
The probability that $\{x_1, x_2, \ldots, x_d, x_{1+d}, \ldots, x_{\ell+d}\}$ is exactly equal to $\mathsf{O}_i^{-1}(y)$, is then

$$\begin{aligned} \frac{\binom{2^n - \alpha_i}{\ell}(2^m - \beta_i - 1)^{2^n - \alpha_i - \ell}}{(2^m - \beta_i)^{2^n - \alpha_i}} &= \frac{\binom{2^n - \alpha_i}{\ell}(2^m - \beta_i - 1)^{2^n - \alpha_i - \ell}}{(2^m - \beta_i)^{\ell}(2^m - \beta_i)^{2^n - \alpha_i - \ell}} \\ &= \binom{2^n - \alpha_i}{\ell} \left(\frac{1}{2^m - \beta_i}\right)^{\ell} \left(1 - \frac{1}{2^m - \beta_i}\right)^{2^n - \alpha_i - \ell}. \end{aligned}$$

□

Now, without loss of generality, assume that whenever the adversary D makes a forward query to O_i, D immediately makes the corresponding backwards query to O^{-1}.Our goal is to show that:

1) All probabilities and conditional probabilities are exactly the same in H_3 as in H_4.
2) The number of remaining choices for oracles O_0 and O_1 is given by Lemma 4.

We will induct on the number q of queries that have already been made.

Case: $q = 0$**.** In H_3 on a forward query, we apply Lemma 5 to obtain the probability

$$\Pr\left[\,\mathsf{O}_i(x) = y_i \text{ and } \mathsf{O}_{1-i}(x) = y_{1-i}\,\right] = \frac{1}{\left(2^m - \beta_i\right)\left(2^m - \beta_{1-i}\right)} = \frac{1}{2^{2m}}.$$

In H_4 on a forward query to O_i, the probability that we output $y_i \xleftarrow{\$} \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_i)$ is equal to

$$\Pr\left[\mathsf{ROra}_4(\mathsf{O}_i, x) = y_i\right] = \frac{1}{2^m - |\mathsf{Im}(\mathsf{InvReg}_i)|} = \frac{1}{2^m}.$$

The $\mathsf{FReact}_4(i,(x,y_i))$ process then simulates the query $\mathsf{O}_{1-i}(x)$, which gives the value y_{1-i} with probability

$$\Pr\left[\mathsf{ROra}_4(\mathsf{O}_{1-i}, x) = y_{1-i}\right] = \frac{1}{2^m - |\mathsf{Im}(\mathsf{InvReg}_{1-i})|} = \frac{1}{2^m}.$$

Thus,

$$\Pr\left[\mathsf{ROra}_4(\mathsf{O}_i, x) = y_i \text{ and } \mathsf{ROra}_4(\mathsf{O}_{1-i}, x) = y_{1-i}\right] = \frac{1}{\left(2^m\right)^2} = \frac{1}{2^{2m}}.$$

In other words, the distributions in H_3 and H_4 on a forward query when $q = 0$ is identical. On a backwards query $\mathsf{O}^{-1}(y)$ in H_3, we wish to find the probability that y has $k \geq 0$ preimages.

There are a total of $\left(2^m\right)^{2^n}$ oracles. There are $\binom{2^n}{k}$ ways for which we can choose k domain elements. This leaves a remaining $2^n - k$ domain elements to be mapped to the remaining $2^m - 1$ codomain elements, for which there are $(2^m - 1)^{2^n - k}$ such oracles. Thus, there are

$$\binom{2^n}{k}\left(2^m - 1\right)^{2^n - k}$$

oracles for which y has k preimages, and therefore

$$\begin{aligned}
\Pr[y \text{ has } k \text{ preimages}] &= \frac{\binom{2^n}{k}\left(2^m - 1\right)^{2^n - k}}{\left(2^m\right)^{2^n}} \\
&= \frac{\binom{2^n}{k}\left(2^m - 1\right)^{2^n - k}}{\left(2^m\right)^{2^n - k}\left(2^m\right)^{k}} \\
&= \binom{2^n}{k}\left(\frac{1}{2^m}\right)^k\left(1 - \frac{1}{2^m}\right)^{2^n - k}.
\end{aligned}$$

In H_4, by construction and the definition of the Binomial distribution, since $\alpha_i = \beta_i = 0$, we have

$$\Pr[y \text{ has } k \text{ preimages}] = \binom{2^n}{k}\left(\frac{1}{2^m}\right)^k\left(1-\frac{1}{2^m}\right)^{2^n-k}.$$

Thus, when $q = 0$, the distributions in H_3 and H_4 are identical.

Case: $q > 0$**.** Now suppose that after q queries, (1) is satisfied so that H_3 and H_4 have the same probabilities and conditional probabilities.

By Lemma 5, on a forward query in H_3, we have

$$\Pr[\mathsf{O}_i(x) = y_i \text{ and } \mathsf{O}_{1-i}(x) = y_{1-i}] = \frac{1}{\big(2^m-\beta_i\big)(2^m-\beta_{1-i})}$$

In H_4 on a forward query to O_i, the probability that we output $y_i \xleftarrow{\$} \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_i)$ is equal to

$$\Pr\left[\mathsf{ROra}_4(\mathsf{O}_i, x) = y_i\right] = \frac{1}{2^m - |\mathsf{Im}(\mathsf{InvReg}_i)|} = \frac{1}{2^m-\beta_i}.$$

The $\mathsf{FReact}_4(i,(x,y_i))$ process then simulates the query $\mathsf{O}_{1-i}(x)$, which gives, in a similar way, the value y_{1-i} with probability

$$\Pr\left[\mathsf{ROra}_4(\mathsf{O}_{1-i}, x) = y_{1-i}\right] = \frac{1}{2^m - |\mathsf{Im}(\mathsf{InvReg}_{1-i})|} = \frac{1}{2^m-\beta_{1-i}}.$$

Thus,

$$\Pr[\mathsf{ROra}_4(\mathsf{O}_i, x) = y_i \text{ and } \mathsf{ROra}_4(\mathsf{O}_{1-i}, x) = y_{1-i}] = \frac{1}{(2^m-\beta_i)(2^m-\beta_{1-i})},$$

showing that the distributions in H_3 and H_4 on a forward query after q queries are identical.

On a backwards query $\mathsf{O}_i^{-1}(y)$ in H_3, for $1 \le \ell \le q$, denote by k_ℓ the number of added domain values in Reg_i in the ℓ-th query. Then we have

$$\alpha_i = k_1 + k_2 + \ldots + k_q$$

elements taken from the domain after q queries. By Lemma 4, the number of oracles consistent with Reg_i and InvReg_i is $(2^m-\beta_i)^{2^n-\alpha_i}$. Since there are a total of $2^n-\alpha_i$ undetermined domain elements, then choosing an additional k_{q+1} domain elements from $\{0,1\}^n \setminus \mathsf{Dom}(\mathsf{Reg}_i)$ can be done in $\binom{2^n-\alpha_i}{k_{q+1}}$ ways. The remaining $2^n-\alpha_i-k_{q+1}$ domain elements must then be mapped to the remaining $2^m-\beta_i-1$ codomain elements, for which there are $(2^m-\beta_i-1)^{2^n-\alpha_i-k_{q+1}}$ such oracles. Thus, there are

$$\binom{2^n-\alpha_i}{k_{q+1}}(2^m-\beta_i-1)^{2^n-\alpha_i-k_{q+1}}$$

oracles for which y has k_{q+1} additional preimages. Thus, the probability that y has

k_{q+1} additional preimages is:

$$\begin{aligned}
&\frac{\binom{2^n-\alpha_i}{k_{q+1}}(2^m-\beta_i-1)^{2^n-\alpha_i-k_{q+1}}}{(2^m-\beta_i)^{2^n-\alpha_i}} \\
&= \frac{\binom{2^n-\alpha_i}{k_{q+1}}(2^m-\beta_i-1)^{2^n-\alpha_i-k_{q+1}}}{(2^m-\beta_i)^{2^n-\alpha_i-k_{q+1}}(2^m-\beta_i)^{k_{q+1}}} \\
&= \binom{2^n-\alpha_i}{k_{q+1}}\left(\frac{1}{2^m-\beta_i}\right)^{k_{q+1}}\left(1-\frac{1}{2^m-\beta_i}\right)^{2^n-\alpha_i-k_{q+1}}.
\end{aligned}$$

In H_4, we have the identical probability by construction using the definition of the Bionomial distribution. Therefore, the distributions in H_3 and H_4 are identical.

- **Hybrid H_5**
  We now make our hybrid *efficient* by aborting whenever we would be required to sample more than n preimages for a single backwards query. Additionally, for a forward query to O_i (and during FReact), rather than sampling $b_i \xleftarrow{\$} \{0,1\}^m \setminus \mathsf{Im}(\mathsf{InvReg}_i)$, we sample $b_i \xleftarrow{\$} \{0,1\}^m$ and abort if $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$.
  - $\mathsf{LOra}_5(a)$: Return $\mathsf{ROra}_5(\mathsf{O}_0, a) + \mathsf{ROra}_5(\mathsf{O}_1, a)$.
  - ROra_5:
    - On queries of the form $\mathsf{O}_i(a)$ for some $i \in \{0,1\}$:
      1. If $(a, b_i) \in \mathsf{Reg}_i$ for some b_i, return b_i.
      2. Else:
         (a) $b_i \xleftarrow{\$} \{0,1\}^m$.
         (b) if $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$, then ABORT.
         (c) $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b_i)\}$.
         (d) Run $\mathsf{FReact}_5(i, (a, b_i))$.
         (e) Return b_i.
    - On queries of the form $\mathsf{O}_i^{-1}(b)$ for some $i \in \{0,1\}$:
      1. If $(S, b) \in \mathsf{InvReg}_i$ for some S, return S.
      2. Else:
         (a) Let $S = \{x_1, x_2, \ldots, x_d\}$ be the set (possibly empty) of all elements x such that $\mathsf{Reg}_i(x) = b$.
         (b) Choose k from $\mathrm{Bin}(2^n - \alpha_i, (2^m - \beta_i)^{-1})$ where $\alpha_i = |\mathsf{Dom}(\mathsf{Reg}_i)|$ and $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$.
         (c) If $k > n$, then ABORT.
         (d) If $k > 0$,
             i. Choose k domain elements $a_1, \ldots, a_k \in \{0,1\}^n \setminus \mathsf{Dom}(\mathsf{Reg}_i)$ uniformly at random, without replacement.
             ii. $S = S \cup \{a_1, \ldots, a_k\}$.
         (e) $\mathsf{InvReg}_i = \mathsf{InvReg}_i \cup \{(S, b)\}$.
         (f) For $a \in S$,
             i. $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b)\}$.
             ii. Run $\mathsf{FReact}_5(i, (a, b))$.
         (g) Output S.

where we define

$\mathsf{FReact}_5(i,(a,b_i))$:

1. If $a \notin \mathsf{FReg}$:
   (a) $b_{1-i} \xleftarrow{\$} \{0,1\}^m$.
   (b) If $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$, then ABORT.
   (c) $\mathsf{Reg}_{1-i} = \mathsf{Reg}_{1-i} \cup \{(a, b_{1-i})\}$.
   (d) $\mathsf{FReg} = \mathsf{FReg} \cup \{a\}$

Lemma 7. *Let $n \in \mathbb{N}$ with $n \geq 10$ and let D be any oracle Turing that makes at most $q \leq \frac{2^n}{n+1}$ queries to its oracles. Then*

$$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_4,\mathsf{ROra}_4}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_5,\mathsf{ROra}_5}(1^n) = 1]\right| \leq \frac{2nq^2}{2^m} + \frac{1}{2^n}$$

Proof. There are three places where the hybrids could differ. For each difference, we will compute the probability that the differing event occurs.

– **Sampling $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$ during a forward query to O_i:** In H_5, on a forward query to O_i, the probability that we sample some $b_i \xleftarrow{\$} \{0,1\}^m$ and end up with $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$ is

$$\Pr[\text{For a fixed query, a sampled } b_i \in \mathsf{Im}(\mathsf{InvReg}_i)] \leq \frac{q}{2^m}$$

since $|\mathsf{Im}(\mathsf{InvReg}_i)| \leq q$ where q is the number of queries. Thus, the probability, that such a situation occurs in any of the q queries is bounded by

$$\Pr[\text{For any query, a sampled } b_i \in \mathsf{Im}(\mathsf{InvReg}_i)] \leq \frac{q^2}{2^m}$$

– **Sampling $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$ during FReact:** Using a similar argument, in H_5, during FReact, the probability that we sample some $b_{1-i} \xleftarrow{\$} \{0,1\}^m$ and end up with $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$ is

$$\Pr[\text{For a fixed query, a sampled } b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})] \leq \frac{q}{2^m}$$

Since each query invokes at most n calls to FReact, the probability that such a situation occurs in any of the q queries is bounded by

$$\Pr[\text{For any query, a sampled } b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})] \leq \frac{nq^2}{2^m}$$

– **Aborting because $k > n$ during a query to O_i^{-1}:**

During a query to O_i^{-1},

$$\Pr[k=\ell] = \Pr[\mathrm{Bin}(2^n-\alpha_i,(2^m-\beta_i)^{-1})=\ell] \tag{2}$$

$$= \binom{2^n-\alpha_i}{\ell}\left(\frac{1}{2^m-\beta_i}\right)^{\ell}\left(1-\frac{1}{2^m-\beta_i}\right)^{2^n-\alpha_i-\ell} \tag{3}$$

$$< \binom{2^n-\alpha_i}{\ell}\left(\frac{1}{2^m-\beta_i}\right)^{\ell}$$

$$\leq \left(\frac{e(2^n-\alpha_i)}{\ell}\right)^{\ell}\left(\frac{1}{2^m-\beta_i}\right)^{\ell} \tag{4}$$

$$\leq \left(\frac{e\cdot 2^n}{\ell}\right)^{\ell}\left(\frac{1}{2^m-\beta_i}\right)^{\ell} \tag{5}$$

$$= \left(\frac{1}{2}\right)^{\ell}\left(\frac{2e\cdot 2^n}{\ell\cdot(2^m-\beta_i)}\right)^{\ell} \tag{6}$$

Since $n \leq m$ and $\beta_i \leq q$, then

$$\Pr[k=\ell] \leq \left(\frac{1}{2}\right)^{\ell}\left(\frac{2e\cdot 2^n}{\ell\cdot(2^n-q)}\right)^{\ell} \tag{7}$$

$$\tag{8}$$

By assumption $n \geq 10$ which implies that $2^{n-1} > \frac{2^n}{n+1} \geq q$ giving us the following inequalities, for $\ell \geq n+1$

$$\ell(2^n-q) \geq 11(2^n-q) \geq 4e(2^n-2^{n-1}) = 2e\cdot 2^n$$

meaning that for $\ell \geq n+1$,

$$\Pr[k=\ell] \leq \left(\frac{1}{2}\right)^{\ell} \tag{9}$$

Summing over all possible values of k between $n+1$ and $2^n-\alpha_i$, we obtain

$$\Pr[k>n] = \sum_{\ell=n+1}^{2^n-\alpha_i}\Pr[k=\ell] \tag{10}$$

$$< \sum_{\ell=n+1}^{2^n-\alpha_i}\frac{1}{2^\ell} \tag{11}$$

$$= \frac{(1/2)^{2^n-\alpha_i+1}-(1/2)^{n+1}}{(1/2)-1}$$

$$= \frac{1}{2^n}-\frac{1}{2^{2^n-\alpha_i}}$$

$$< \frac{1}{2^n}.$$

Thus, by the union bound, the probability that any of these differences occur is

$$\leq \frac{q^2}{2^m}+\frac{nq^2}{2^m}+\frac{1}{2^n} \leq \frac{2nq^2}{2^m}+\frac{1}{2^n}.$$

□

- **Hybrid H_6**
  In this hybrid, we change FReact so that it sets the reactionary query $\mathsf{O}_{1-i}(a)$ equal to $R(a) - \mathsf{O}_i(a)$ where R is a truly random oracle.

  – $\mathsf{LOra}_6(a)$: Same as previous hybrid.

  – ROra_6: Same as previous hybrid.

  $\mathsf{FReact}_6(i, (a, b_i))$:

  1. If $a \notin \mathsf{FReg}$, for any y:
     (a) Set $b_{1-i} = R(a) - b_i$.
     (b) If $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$, then ABORT.
     (c) $\mathsf{Reg}_{1-i} = \mathsf{Reg}_{1-i} \cup \{(a, b_{1-i})\}$.
     (d) $\mathsf{FReg} = \mathsf{FReg} \cup \{a\}$

  Lemma 8. *For all $n \in \mathbb{N}$ and all oracle Turing Machines D,*

  $$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_5, \mathsf{ROra}_5}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_6, \mathsf{ROra}_6}(1^n) = 1]\right| = 0.$$

  Proof. By construction, we have ensured that $R(a)$ is only ever queried once on each a. Thus, the distribution of $R(a) - b_i$ is uniform and the hybrids are identical. □

- **Hybrid H_7**
  We now answer queries to LOra using a truly random oracle R. Additionally, we store queries to LOra in LReg and abort if during a query to O_i^{-1}, we sample a preimage $a_i \in \mathsf{LReg} \setminus \mathsf{FReg}$. Furthermore, in H_7 and onward, FReact_7 becomes crucial. Since we now define $\mathsf{LOra}(a)$ via an independent function $R(a)$ rather than as the sum $\mathsf{O}_0(a) + \mathsf{O}_1(a)$, a query to $\mathsf{LOra}(a)$ followed by queries to $\mathsf{O}_0(a)$ and $\mathsf{O}_1(a)$ could otherwise produce inconsistent outputs. The sub-process FReact_7 ensures that the component oracles remain synchronized and that the overall transcript is consistent.

  – $\mathsf{LOra}_7(a)$:
    1. $\mathsf{LReg} = \mathsf{LReg} \cup \{a\}$.
    2. Compute $\mathsf{ROra}_7(\mathsf{O}_0, a) + \mathsf{ROra}_7(\mathsf{O}_1, a)$.
    3. Return $R(a)$.

  – ROra_7:
    * On queries of the form $\mathsf{O}_i(a)$ for some $i \in \{0, 1\}$:
      1. If $(a, b_i) \in \mathsf{Reg}_i$ for some b_i, return b_i.
      2. Else:
         (a) $b_i \xleftarrow{\$} \{0,1\}^m$.
         (b) if $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$, then ABORT.
         (c) $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b_i)\}$.
         (d) Run $\mathsf{FReact}_7(i, (a, b_i))$.
         (e) Return b_i.
    * On queries of the form $\mathsf{O}_i^{-1}(b)$ for some $i \in \{0, 1\}$:
      1. If $(S, b) \in \mathsf{InvReg}_i$ for some S, return S.
      2. Else:
         (a) Let $S = \{x_1, x_2, \ldots, x_d\}$ be the set (possibly empty) of all elements x such that $\mathsf{Reg}_i(x) = b$.

(b) Choose k from $\mathrm{Bin}(2^n - \alpha_i, (2^m - \beta_i)^{-1})$ where $\alpha_i = |\mathsf{Dom}(\mathsf{Reg}_i)|$ and $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$.
(c) If $k > n$, then ABORT.
(d) If $k > 0$,
  i. Choose k domain elements $a_1, \ldots, a_k \in \{0,1\}^n \setminus \mathsf{Dom}(\mathsf{Reg}_i)$ uniformly at random, without replacement.
  ii. If any $a_i \in \mathsf{LReg} \setminus \mathsf{FReg}$, ABORT.
  iii. $S = S \cup \{a_1, \ldots, a_k\}$.
(e) $\mathsf{InvReg}_i = \mathsf{InvReg}_i \cup \{(S, b)\}$.
(f) For $a \in S$,
  i. $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b)\}$.
  ii. Run $\mathsf{FReact}_7(i, (a, b))$.
(g) Output S.

where we define

$\mathsf{FReact}_7(i, (a, b_i))$:

1. If $a \notin \mathsf{FReg}$:
   (a) Set $b_{1-i} = R(a) - b_i$.
   (b) If $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$, then ABORT.
   (c) $\mathsf{Reg}_{1-i} = \mathsf{Reg}_{1-i} \cup \{(a, b_{1-i})\}$.
   (d) $\mathsf{FReg} = \mathsf{FReg} \cup \{a\}$

Lemma 9. *For all $n \in \mathbb{N}$ and all oracle Turing Machines D,*

$$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_6, \mathsf{ROra}_6}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_7, \mathsf{ROra}_7}(1^n) = 1]\right| = 0.$$

Proof. By construction, we have ensured that $R(a) = \mathsf{ROra}_7(\mathsf{O}_0, a) + \mathsf{ROra}_7(\mathsf{O}_1, a)$. Additionally, the new abort condition will never occur. Thus, the hybrids have identical behavior. □

- **Hybrid H_8**
  We no longer query the right oracle ROra during queries to the left oracle LOra.
  - $\mathsf{LOra}_8(a)$:
    1. $\mathsf{LReg} = \mathsf{LReg} \cup \{a\}$.
    2. ~~Compute $\mathsf{ROra}_8(\mathsf{O}_0, a) + \mathsf{ROra}_8(\mathsf{O}_1, a)$.~~
    3. Return $R(a)$ where R is a random oracle.
  - ROra_8:
    * On queries of the form $\mathsf{O}_i(a)$ for some $i \in \{0,1\}$:
      1. If $(a, b_i) \in \mathsf{Reg}_i$ for some b_i, return b_i.
      2. Else:
         (a) $b_i \xleftarrow{\$} \{0,1\}^m$.
         (b) if $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$, then ABORT.
         (c) $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b_i)\}$.
         (d) Run $\mathsf{FReact}_8(i, (a, b_i))$.
         (e) Return b_i.
    * On queries of the form $\mathsf{O}_i^{-1}(b)$ for some $i \in \{0,1\}$:

1. If $(S, b) \in \mathsf{InvReg}_i$ for some S, return S.
2. Else:
   (a) Let $S = \{x_1, x_2, \ldots, x_d\}$ be the set (possibly empty) of all elements x such that $\mathsf{Reg}_i(x) = b$.
   (b) Choose k from $\mathrm{Bin}(2^n - \alpha_i, (2^m - \beta_i)^{-1})$ where $\alpha_i = |\mathsf{Dom}(\mathsf{Reg}_i)|$ and $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$.
   (c) If $k > n$, then ABORT.
   (d) If $k > 0$,
      i. Choose k domain elements $a_1, \ldots, a_k \in \{0,1\}^n \setminus \mathsf{Dom}(\mathsf{Reg}_i)$ uniformly at random, without replacement.
      ii. If any $a_i \in \mathsf{LReg} \setminus \mathsf{FReg}$, ABORT.
      iii. $S = S \cup \{a_1, \ldots, a_k\}$.
   (e) $\mathsf{InvReg}_i = \mathsf{InvReg}_i \cup \{(S, b)\}$.
   (f) For $a \in S$,
      i. $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b)\}$.
      ii. Run $\mathsf{FReact}_8(i, (a, b))$.
   (g) Output S.

where we define

$\mathsf{FReact}_8(i, (a, b_i))$:

1. If $a \notin \mathsf{FReg}$:
   (a) Set $b_{1-i} = R(a) - b_i$.
   (b) If $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$, then ABORT.
   (c) $\mathsf{Reg}_{1-i} = \mathsf{Reg}_{1-i} \cup \{(a, b_{1-i})\}$.
   (d) $\mathsf{FReg} = \mathsf{FReg} \cup \{a\}$

Lemma 10. *Let $n \in \mathbb{N}$ with $n \geq 10$ and let D be any oracle Turing that makes at most $q \leq \frac{2^n}{n+1}$ queries to its oracles. Then,*

$$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_7, \mathsf{ROra}_7}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_8, \mathsf{ROra}_8}(1^n) = 1]\right| \leq \frac{6nq^2}{2^m} + \frac{4n^2q^2}{2^n} + \frac{2}{2^n}.$$

Proof. Merely sending additional forward queries to $\mathsf{ROra}(\mathsf{O}_i, \cdot)$ does not affect the output distribution of other queries to LOra or ROra, apart from increasing the probability of aborting. Thus, conditioned on the hybrids not aborting, the hybrids are identically distributed.

We now consider the probability of aborting in each of the abort cases. The following analysis holds for both hybrids:

– **Aborting during a forward query to O_i:** We only abort during a forward query to $\mathsf{O}_i(a)$ if we sample $b_i \leftarrow \{0,1\}^m$ and have either $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$ or $b_{1-i} = R(a) - b_i \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$. Since $|\mathsf{Im}(\mathsf{InvReg}_i)| \leq q$ and $|\mathsf{Im}(\mathsf{InvReg}_{1-i})| \leq q$, then for any fixed value of $R(a)$,

$$\Pr[\text{For a fixed query, a sampled } b_i \in \mathsf{Im}(\mathsf{InvReg}_i) \text{ or } R(a) - b_i \in \mathsf{Im}(\mathsf{InvReg}_{1-i})] \leq \frac{2q}{2^m}$$

Thus, the probability, that such a situation occurs in any of the q queries is bounded by

$$\Pr[\text{For any query, a sampled } b_i \in \mathsf{Im}(\mathsf{InvReg}_i) \text{ or } R(a) - b_i \in \mathsf{Im}(\mathsf{InvReg}_{1-i})] \leq \frac{2q^2}{2^m}$$

– **Aborting because $k > n$ during a query to O_i^{-1}:** By a similar argument as in Lemma 7, the probability of this abort occurring is $\leq \frac{1}{2^n}$.

– **Aborting because $a_i \in \mathsf{LReg} \backslash \mathsf{FReg}$ during a query to O_i^{-1}:** Since $|\mathsf{LReg}| \leq q$ and $|\mathsf{Dom}(\mathsf{Reg}_i)| \leq nq$, then the probability that a uniformly sampled $a_i \leftarrow \{0,1\}^n \setminus \mathsf{Dom}(\mathsf{Reg}_i)$ ends up in $\mathsf{LReg} \setminus \mathsf{FReg}$ is $\leq \frac{q}{2^n - nq}$. Since we sample at most nq such a_i during the hybrids, the probability that any sampled a_i causes this abort is at most

$$\frac{nq^2}{2^n - nq} \leq \frac{nq^2}{2^n - n\left(\frac{2^n}{n+1}\right)} \tag{12}$$

$$= \frac{nq^2}{\left(\frac{2^n}{n+1}\right)(n+1-n)} \tag{13}$$

$$= \frac{n(n+1)q^2}{2^n}$$

$$\leq \frac{2n^2q^2}{2^n}.$$

For $q \leq \frac{2^n}{n+1}$

– **Aborting because $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$ while running FReact on a backwards query to O_i^{-1}.** Observe that since we abort if $a_i \in \mathsf{LReg} \backslash \mathsf{FReg}$, then we will only sample b_{1-i} during a backwards query to O_i^{-1} if $R(a)$ has not previously been queried. Thus, $b_{1-i} = R(a) - b_i$ is uniformly random. Therefore, by a similar argument as in Lemma 7, the probability of this abort occurring is $\leq \frac{nq^2}{2^m}$.

□

Thus, the probability that either hybrid would experience an abort is

$$\leq 2\left(\frac{2q^2}{2^m} + \frac{nq^2}{2^m} + \frac{2n^2q^2}{2^n} + \frac{1}{2^n}\right) \leq \frac{6nq^2}{2^m} + \frac{4n^2q^2}{2^n} + \frac{2}{2^n}$$

- **Hybrid H_9**
  In our final hybrid, we remove the LReg register and the corresponding abort condition. Note that we are now in the ideal world since ROra can be simulated by a polynomial time simulator Sim^R with oracle access to R.

  – $\mathsf{LOra}_9(a)$: Return $R(a)$ where R is a random oracle.

  – ROra_9: The simulator Sim^R does the following:

    * On queries of the form $\mathsf{O}_i(a)$ for some $i \in \{0,1\}$:
      1. If $(a, b_i) \in \mathsf{Reg}_i$ for some b_i, return b_i.
      2. Else:
         (a) $b_i \xleftarrow{\$} \{0,1\}^m$.
         (b) if $b_i \in \mathsf{Im}(\mathsf{InvReg}_i)$, then ABORT.
         (c) $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a, b_i)\}$.

(d) Run $\mathsf{FReact}_9(i,(a,b_i))$.

(e) Return b_i.

$*$ On queries of the form $\mathsf{O}_i^{-1}(b)$ for some $i \in \{0,1\}$:

1. If $(S,b) \in \mathsf{InvReg}_i$ for some S, return S.
2. Else:
   (a) Let $S = \{x_1, x_2, \ldots, x_d\}$ be the set (possibly empty) of all elements x such that $\mathsf{Reg}_i(x) = b$.
   (b) Choose k from $\mathrm{Bin}(2^n - \alpha_i, (2^m - \beta_i)^{-1})$ where $\alpha_i = |\mathsf{Dom}(\mathsf{Reg}_i)|$ and $\beta_i = |\mathsf{Im}(\mathsf{InvReg}_i)|$.
   (c) If $k > n$, then ABORT.
   (d) If $k > 0$,
      i. Choose k domain elements $a_1, \ldots, a_k \in \{0,1\}^n \setminus \mathsf{Dom}(\mathsf{Reg}_i)$ uniformly at random, without replacement.
      ii. ~~If any $a_i \in \mathsf{LReg} \setminus \mathsf{FReg}$, ABORT.~~
      iii. $S = S \cup \{a_1, \ldots, a_k\}$.
   (e) $\mathsf{InvReg}_i = \mathsf{InvReg}_i \cup \{(S,b)\}$.
   (f) For $a \in S$,
      i. $\mathsf{Reg}_i = \mathsf{Reg}_i \cup \{(a,b)\}$.
      ii. Run $\mathsf{FReact}_9(i,(a,b))$.
   (g) Output S.

where we define

$\mathsf{FReact}_9(i,(a,b_i))$:

1. If $a \notin \mathsf{FReg}$:
   (a) Set $b_{1-i} = R(a) - b_i$.
   (b) If $b_{1-i} \in \mathsf{Im}(\mathsf{InvReg}_{1-i})$, then ABORT.
   (c) $\mathsf{Reg}_{1-i} = \mathsf{Reg}_{1-i} \cup \{(a, b_{1-i})\}$.
   (d) $\mathsf{FReg} = \mathsf{FReg} \cup \{a\}$

Lemma 11. *Let $n \in \mathbb{N}$ with $n \geq 10$ and let D be any oracle Turing that makes at most $q \leq \frac{2^n}{n+1}$ queries to its oracles. Then,*

$$\left|\Pr[\mathsf{D}^{\mathsf{LOra}_8,\mathsf{ROra}_8}(1^n) = 1] - \Pr[\mathsf{D}^{\mathsf{LOra}_9,\mathsf{ROra}_9}(1^n) = 1]\right| \leq \frac{2n^2q^2}{2^n}.$$

Proof. The only difference between the hybrids is that we have removed the abort condition for when $a_i \in \mathsf{LReg} \setminus \mathsf{FReg}$. As shown in Lemma 10, the probability that we would abort in hybrid H_8 due to sampling some $a_i \in \mathsf{LReg} \setminus \mathsf{FReg}$ is $\leq \frac{2n^2q^2}{2^n}$. Thus, this is the distinguishing probability. □

By combining our hybrid indistinguishability lemmas together, we get that the distin-

guishing advantage between the real world and ideal world is

$$\begin{aligned}
&\Pr[\left|\mathsf{D}^{\mathsf{LOra}_1,\mathsf{ROra}_1}(1^n)=1\right] - \Pr[\mathsf{D}^{\mathsf{LOra}_9,\mathsf{ROra}_9}(1^n)=1]\right| \\
&\leq \left(\frac{2nq^2}{2^m}+\frac{1}{2^n}\right)+\left(\frac{6nq^2}{2^m}+\frac{4n^2q^2}{2^n}+\frac{2}{2^n}\right)+\frac{2n^2q^2}{2^n} \\
&\leq \frac{8nq^2}{2^m}+\frac{6n^2q^2}{2^n}+\frac{3}{2^n} \\
&\leq \frac{8nq^2}{2^m}+\frac{7n^2q^2}{2^n}.
\end{aligned}$$

□

References

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. arXiv:https://doi.org/10.1137/0213053, doi:10.1137/0213053.

[BN18] Srimanta Bhattacharya and Mridul Nandi. Full indifferentiable security of the xor of two or more random permutations using the χ^2 method. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 387–412, April / May 2018. doi:10.1007/978-3-319-78381-9_15.

[DKT16] Dana Dachman-Soled, Jonathan Katz, and Aishwarya Thiruvengadam. 10-round Feistel is indifferentiable from an ideal cipher. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 649–678, May 2016. doi:10.1007/978-3-662-49896-5_23.

[GBJ+23] Aldo Gunsing, Ritam Bhaumik, Ashwin Jha, Bart Mennink, and Yaobin Shen. Revisiting the indifferentiability of the sum of permutations. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 628–660, August 2023. doi:10.1007/978-3-031-38548-3_21.

[GM19] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali*, pages 173–201. 2019.

[Gol01] Oded Goldreich. *Foundations of Cryptology: Basic Tools*. Cambridge, 2001.

[GS23] Riddhi Ghosal and Amit Sahai. Building hard problems by combining easy ones. In *IEEE International Symposium on Information Theory, ISIT 2023, Taipei, Taiwan, June 25-30, 2023*, pages 1770–1775. IEEE, 2023. doi:10.1109/ISIT54713.2023.10206689.

[Gun22] Aldo Gunsing. Block-cipher-based tree hashing. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 205–233, August 2022. doi:10.1007/978-3-031-15985-5_8.

[MP15] Bart Mennink and Bart Preneel. On the XOR of multiple random permutations. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *ACNS 15International Conference on Applied Cryptography and Network Security*, volume 9092 of *LNCS*, pages 619–634, June 2015. doi:10.1007/978-3-319-28166-7_30.

[MPN10] Avradip Mandal, Jacques Patarin, and Valérie Nachef. Indifferentiability beyond the birthday bound for the xor of two public random permutations. In Guang Gong and Kishan Chand Gupta, editors, *INDOCRYPT 2010*, volume 6498 of *LNCS*, pages 69–81, December 2010.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39, February 2004. doi:10.1007/978-3-540-24638-1_2.

[Sha49] Claude E Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949.

[Yao82] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.