Black-Box Composition Does Not Imply Adaptive Security

Steven Myers*

Department of Computer Science
University of Toronto
Canada

Abstract. In trying to provide formal evidence that composition has security increasing properties, we ask if the composition of non-adaptively secure permutation generators necessarily produces adaptively secure generators. We show the existence of oracles relative to which there are non-adaptively secure permutation generators, but where the composition of such generators fail to achieve security against adaptive adversaries. Thus, any proof of security for such a construction would need to be non-relativizing. This result can be used to partially justify the lack of formal evidence we have that composition increases security, even though it is a belief shared by many cryptographers.

1 Introduction

While there is arguably no strong theory that guides the development of block-ciphers such as DES and AES, there is a definite belief in the community that the composition of functions often results in functions that have stronger security properties than their constituents. This is evident as many ciphers such as DES, AES and MARS have a "round structure" at the heart of their constructions, and a large part of the ciphers' apparent security comes from the composition of these rounds.

In an attempt to understand the security benefits of composition, there have been several papers that have tried to quantify different ways in which the composition of functions increases security properties as compared to the constituent functions [14, 1]. A natural question along these lines is to look at functions that are pseudo-random from the perspective of a non-adaptive adversary, but not that of the standard adaptive adversary, and ask if composition of these functions necessarily provides security against adaptive adversaries. It appears that at least some people in the cryptographic community believe this to be true. In fact, recently Maurer and Pietrzak [16] have shown the cascade of two non-adaptively *statistically-secure* permutations results in an adaptively secure construction, where the cascade of two generators is the composition of the first with the inverse of the second. Additionally, they ask if their cascade construction can be proven secure in the computational setting.

* E-Mail: myers@cs.toronto.edu

In this paper we show that there is no non-relativizing proof that composition of functions provides security against adaptive adversaries. Thus, this work falls into a general research program that demonstrates the limitations of black-box constructions in cryptography. Examples of such research include [12, 19, 13, 5, 7, 6]. In the final section, we discuss how the techniques used here can be lifted and used on at least one other natural construction: the XOR of function generators.

We note that it is not possible to strictly separate non-adaptively secure function generators from adaptively secure ones in the black-box model, as there are several black-box constructions that construct the stronger object from the weaker one. The first treats the non-adaptively secure generator as a pseudo-random number generator and then uses the construction of Goldreich, Goldwasser and Micali [9] in order to construct a pseudo-random function generator. The second construction treats the non-adaptively secure function generator as a synthesizer and then constructs a function generator as described by Naor and Reingold in [17]. In both cases, we can go from function generators to permutation generators through the well known Luby-Rackoff construction [15]. However, there are several reasons why these constructions are unsatisfying: first, these constructions are not representative of what is done in practice to construct block-ciphers; second, they require $\Omega(\frac{n}{\log n})$ calls to the non-adaptively secure functions generators. Therefore it is natural to ask if the more efficient constructions used in practice can provide adaptive security.

Finally, since it is possible to construct adaptively secure generators from non-adaptively secure generators using black box techniques, this result suggests the possibility that one reason there may be few general theorems championing the general security amplification properties of compositions is that such theorems are not establishable using standard black-box proof techniques.

1.1 Black-Box Constructions and Proofs

Since the existence of most modern cryptographic primitives imply $\mathcal{P} \neq \mathcal{NP}$, much of modern cryptography revolves around trying to construct more complex primitives from other simpler primitives that are assumed to exist. That is, if we assume primitives of type P exist, and wish to show that a primitive of type Q exists, then we give a construction C, where $C(M_P)$ is an implementation of Q whenever M_P is an implementation of P. However, most constructions in modern cryptography are black-box. More specifically, when given a a primitive P, we construct a primitive Q by a construction C^P, where the primitive P is treated as an oracle. The difference between the two constructions is that in the former case the construction may make use of the machine description, while in the latter it only treats the primitive as an oracle to be queried: it's as if P were inside of a black box.

Observe that it is not immediately clear how to prove that there can be no black-box construction C^P of a primitive Q from an implementation M_P of a primitive P, as the implementation C and the proof of its correctness and security could always ignore the presence of the oracle P, and independently use the implementation M_P in the construction C. The notion of proving black-box separation results was initiated by Baker, Gill and Solovay [2], who were interested in the techniques necessary to answer the $\mathcal{P}$ vs. $\mathcal{NP}$ question. Building on this work, Impagliazzo and Rudich [12] gave a model in which one can prove separations for cryptographic primitives. In their model they note that black-box constructions and proofs work relative to any oracle, that is they relativize, and therefore it is sufficient to provide an oracle O which implements a primitive P, but all constructions C^O of primitive Q are not secure relative to O. Gertner, Malkin and Reingold [8] have shown that if one's goal is to rule out black-box constructions, then a weaker type of theorem will suffice: for each black-box construction C^P of primitive Q, it suffices to demonstrate an oracle O that implements primitive P, but for which C^O is insecure. Our result will be of this flavor.

As was stated previously, we cannot separate non-adaptive generators from adaptive ones, as there are black-box constructions of one from the other. However, we show that certain constructions (those which are the composition of permutation generators) cannot provide provable adaptive security using black-box techniques. This is done by constructing an oracle for each construction that provides a natural representation of a non-adaptively secure permutation generator, but where the composition of these generators is not adaptively secure.

Finally, we note that there are several techniques that are used in cryptography, such as Zero-Knowledge in its many incarnations (to name but a few [11, 10, 4, 18, 3]) that are often used in cryptographic constructions in such a way that the construction, and not necessarily the technique, is non-black-box.

1.2 Our Results

Our main result involves permutation generators. These generators have an associated domain-size parameter $n \in \mathbb{N}$ that fixes the set $\{0,1\}^n$ over which the permutations are defined.

Theorem 1. *For every polynomial m, there exists a pair of oracles relative to which there exist non-adaptively secure pseudo-random permutation generators P such that the generator $\underbrace{P \circ \ldots \circ P}_{m(n)}$ is not adaptively secure.*

In the theorem $P \circ P'$ denotes the natural composition construction: it is the generator constructed by fixing the security parameter and randomly choosing a $p \in P$ and $p' \in P'$ and computing the permutation $p \circ p'$. The construction $\underbrace{P \circ \ldots \circ P}_{m(n)}$ defines the generator that generates permutations over the set $\{0,1\}^n$ by composing $m(n)$ generators P.

1.3 Preliminaries & Notations

Let S be a finite set, and let $x \in_{\mathcal{U}} S$ denote the act of choosing an element x uniformly at random from S. To describe some of the probabilistic experiments we adopt the notation $\Pr[R_1; ...; R_k :: E|C]$ to denote the probability that if random processes R_1 through R_k are performed, in order, then, conditioned on event C, the event E occurs.

Notation 1. *Let D be any finite set, and let $Q_1, Q_2 \subseteq D \times D$ be arbitrary sets of pairs. We define $Q_1 \circ Q_2 = \{(a, c) | \exists a, b, c \in D \text{ s.t. } (a, b) \in Q_1 \wedge (b, c) \in Q_2\}$. Generalizing this notion, for a collection of sets $Q_{k_1}, \ldots, Q_{k_m} \subseteq D \times D$ and for a vector $\boldsymbol{K} = (k_1, ..., k_m)$, let $Q_{\boldsymbol{K}} = Q_{k_1} \circ \ldots \circ Q_{k_m}$.*

Notation 2. *For any finite set D, we denote by $D^{[k]}$ the set of all k-tuples of* distinct *elements from D. In a slight abuse of notation, for a k-tuple $\boldsymbol{d} = (d_1, \ldots, d_k) \in D^k$, we say $x \in \boldsymbol{d}$ if there exists an $i \leq k$ such that $x = d_i$. Additionally, for a function f of the form $D \to D$, we write $f(\boldsymbol{d})$ to denote $(f(d_1), \ldots, f(d_k))$.*

Notation 3. *We denote by Π^n the set of all permutations over $\{0,1\}^n$, and we denote by $\mathcal{F}^n$ the set off all functions of the form $\{0,1\}^n \to \{0,1\}^n$.*

1.4 Organization

In Section 2 we introduce the standard definitions related to Pseudo-Random Permutation and Function Generators, the difference between adaptive and non-adaptive security, and we discuss how these definitions are lifted into relativized worlds. In Section 3 we present the oracles relative to which we will prove our result. We show that, relative to these oracles, non-adaptively secure permutation generators exist, but that their composition does not provide adaptive security. This is done by showing that non-adaptive adversaries cannot make effective use of one of the oracles that an adaptive adversary can make use of. We demonstrate the oracles' lack of effectiveness to the non-adaptive adversary by demonstrating how the oracles responses could easily be simulated by a non-adaptive adversary. In Section 4 we present the proofs of the combinatorial lemmas behind the simulation just mentioned. We finish in Section 5 by discussing how the techniques presented can be lifted to get similar results for other constructions, such as those based on XOR. Finally, we discuss some directions for future work.

2 Standard Definitions

We use the standard, Turing machine based, uniform definitions for pseudo-random function generators and adversaries.

Definition 1 (Function Ensembles). *We call $G : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ a function generator. We say that $k \in \{0,1\}^\kappa$ is a* key *of G, write $G(k, \cdot)$ as $g_k(\cdot)$ and say that key k chooses the function g_k. Let $g \in_{\mathcal{U}} G$ represent the act of uniformly at random choosing a key k from $\{0,1\}^\kappa$, and then using the key k to choose the function g_k.*

Let ℓ be a polynomial, and let $\mathcal{N} \subseteq \mathbb{N}$ be an infinitely large set. For each $n \in \mathcal{N}$, let $G^n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^n$ be a function generator.

We call $G = \{G^n | n \in \mathcal{N}\}$ *a* function ensemble. *Given an ensemble* G*, if for every* $n \in \mathcal{N}$*, the function* $g \in_{\mathcal{U}} G^n$ *is a permutation, then we say* G *is a* permutation ensemble. *We say an ensemble* G *is* efficiently computable *if there exists a Turing machine* M *and a polynomial* p *such that for all sufficiently large* n*, for all* $x \in \{0,1\}^n$ *and* $k \in \{0,1\}^{\ell(n)}$ *the Turing machine's* $M(k,x)$ *output is* $G^n_k(x)$ *and* $M(k,x)$ *runs in time* $p(n)$*.*

Definition 2 ((Non-)Adaptive Adversaries). *An adversary,* A*, is a probabilistic, polynomial-time Turing machine with oracle access that outputs an element in* $\{0,1\}$*. We denote an adversary* A *with access to an oracle* f *as* A^f*. In order to query an oracle* f*,* A *writes its query to a special oracle-query-tape, and enters a specified query request state. The response to the query is then written to an oracle-response-tape by the oracle, and* A *continues its computation. For accounting purposes, we assume that it takes unit time to write the response of the oracle to the tape, once* A *has entered the query state. An adversary is* adaptive *if it can make multiple queries to the oracle, where future queries can depend on the results of previous queries. A* non-adaptive *adversary may make multiple queries to the oracle, but all queries must be made in parallel at the same time. Formally, the adversary is permitted to write several queries at a time to the oracle-query-tape. When the machine enters the specified query state, the response to all of the queries are written to the response tape.*

Definition 3 ((Non-)Adaptive Pseudo-Random Function Generator Ensembles). *Let* m *and* ℓ *be polynomials. Let* $G = \{G^n | n \in \mathbb{N}\}$ *be an efficiently computable function generator ensemble such that for each* n *the generator* G^n *is of the form* $\{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}^{m(n)}$*. Define* $\mathcal{F} = \{\mathcal{F}^n | n \in \mathbb{N}\}$*.*

We say that G *is adaptively (resp. non-adaptively) secure if for all constants* $c > 0$*, for all adaptive (resp. non-adaptive) polynomial time adversaries* A *and for all sufficiently large* n*:*

$$\left| \Pr_{\substack{g \in_{\mathcal{U}} G^n \\ r \in_{\mathcal{U}} \{0,1\}^*}} [A^g(1^n) = 1] - \Pr_{\substack{f \in_{\mathcal{U}} \mathcal{F}^n \\ r \in_{\mathcal{U}} \{0,1\}^*}} [A^f(1^n) = 1] \right| \leq \frac{1}{n^c},$$

where the $r \in \{0,1\}^*$ *represent the random coin-tosses made by* A*.*

In this work we are concerned with the above definitions, but in worlds where a pair of oracles (O, R) exist. We note we use a pair of oracles, as opposed to just one, to simplify the presentation of the proof. We extend the definitions of function ensembles and adaptive/non-adaptive adversaries by allowing Turing machines to have access to the oracles O and R. We stress that non-adaptive adversaries *are* permitted to query O and R in an adaptive manner: the non-adaptive restriction on oracle queries in the definition of the adversary (Defn. 2) are only for the oracles f and g specified in the definition of pseudo-random function generator ensembles (Defn. 3).

3 The Separating Oracles for Composition

We will construct an oracle that contains an information theoretically secure pseudo-random permutation generator (PRPG). By this we mean that for each n it will include 2^n random permutations over $\{0,1\}^n$. Clearly, a non-adaptively secure PRPG F can be immediately constructed from such an oracle, but it is also clear that the same generator will be adaptively secure. Therefore, we add another oracle R that weakens the security of O. To help describe R, suppose the construction of interest is the composition of two permutations from O, and suppose the adversary has access to a function g that is either a function chosen randomly from $\mathcal{F}^{n \to n}$ or $\pi_1 \circ \pi_2$ for $\pi_2, \pi_1 \in_{\mathcal{U}} \Pi^n$. The oracle R iteratively requests the values of $y_i = g(x_i)$ for enough (but still a small number of) randomly chosen values x_i that it should be able to uniquely identify $\pi_1, \pi_2 \in O$, if it is the case that $g = \pi_1 \circ \pi_2$. If R determines that that there exists a $\pi_1, \pi_2 \in O$ such that $y_i = \pi_1 \circ \pi_2(x_i)$ for each i, then it will predict a random input/output pair (x^*, y^*), where $y^* = \pi_1 \circ \pi_2(x^*)$. Alternatively, if there is no pair of permutations in O whose composition is consistent with all of the (x_i, y_i) then the oracle rejects and outputs $\perp$.

The oracle R provides a trivial way for an adaptive adversary to break the security of the composed generators: such an adversary can easily supply the $y_i = g(x_i)$ values R requests as responses to its x_i challenges. If R returns a prediction (x^*, y^*) that is consistent with $y^* = g(x^*)$ then almost surely g is a composition of permutations from O. In contrast, if the adversary is *non-adaptive* then the oracle R will be of essentially no use to the adversary because of R's iterative nature. Therefore, it is as if R does not exist to the adversary, and therefore the adversary cannot use R to help identify permutations that are in O.

3.1 Oracle Definitions

Definition 4 (The Oracle O). *Let $O^n \stackrel{O}{\leftarrow} \Pi^n$ denote the process of choosing an indexed set of 2^n random permutations from Π^n with replacement. Let O^n_k denote the kth permutation in O^n. Let $O = \{O^n | n \in \mathbb{N}\}$. Where n is clear we write O_k to denote O^n_k. For $k_1, ..., k_m \in \{0,1\}^n$ and $\boldsymbol{K} = (k_1, ..., k_m)$, let $O_{\boldsymbol{K}}$ denote $O_{k_m} \circ \ldots \circ O_{k_1}$. Further, for $x_1, ..., x_\ell \in \{0,1\}^n$ and $\boldsymbol{x} = (x_1, ..., x_\ell)$, denote $O_k(\boldsymbol{x}) = \boldsymbol{y} = (O_k(x_1), \ldots, O_k(x_\ell))$.*

Definition 5 (Composition Construction). *Let $m : \mathbb{N} \to \mathbb{N}$ be a polynomial where for every $i \in \mathbb{N}$, $m(i) > 2$. For an oracle O, for every $n \in \mathbb{N}$ and $k_1, ..., k_{m(n)} \in \{0,1\}^n$ let $F^n_{(k_1,\ldots,k_{m(n)})}(x) = O_{k_1,\ldots k_{m(n)}}(x)$. Let $F = \cup_n \{F^n\}$ be the proposed construction for an adaptively secure PRPG.*

Definition 6 (The Oracle R). *For an oracle O as described in Definition 4 and a construction F of m compositions as described in Definition 5 we define the oracle R as follows. Define, with foresight, $\ell(n) = m(n) + 1$. Let $R = \{(R_1, R_2, R_3)\}$ be an oracle that for each n is chosen randomly according to the random process $\Psi^n(O)$, and the fixed. The process $\Psi^n(0)$ is described below:*

$R_1(1^n) \to x_1, \ldots, x_{\ell(n)}$ *where* $(x_1, \ldots, x_{\ell(n)}) \in_{\mathcal{U}} (\{0,1\}^n)^{[\ell(n)]}$.
$R_2(1^n, x_1, \ldots, x_{\ell(n)}, y_1, ..., y_{\ell(n)}) \to x_{\ell(n)+1}, x_{\ell(n)+2}$ *for the* $x_1, \ldots, x_{\ell(n)}$ *output by* $R_1(1^n)$*; any* $y_1, \ldots, y_{\ell(n)} \in \{0,1\}^n$*; and* $\left(x_{\ell(n)+1}, x_{\ell(n)+2}\right) \in_{\mathcal{U}} \left(\{0,1\}^n \setminus \{x_1, \ldots, x_{\ell(n)}\}\right)^{[2]}$.
$R_3(1^n, x_1, \ldots, x_{\ell(n)+2}, y_1, \ldots, y_{\ell(n)+2}) = (x^*, y^*)$ *for the* $(x^{\ell(n)+1}, x^{\ell(n)+2})$ *output by* $R_2(1^n, x_1, \ldots, x_{\ell(n)}, y_1, \ldots, y_{\ell(n)})$*; any* $y_{\ell(n)+1}, y_{\ell(n)+2} \in \{0,1\}^n$*;* $\kappa \in_{\mathcal{U}} \left\{\kappa = (k_1, \ldots, k_{m(n)}) \in \{0,1\}^{n \cdot m(n)} | O_\kappa(x_1, \ldots, x_{\ell(n)+2}) = (y_1, \ldots, y_{\ell(n)+2})\right\}$*;* $x^* \in_{\mathcal{U}} \{0,1\}^n$*; and* $y^* = O_\kappa(x^*)$.

On all other inputs to the oracles R_1, R_2 *and* R_3 *the result is* $\perp$*. Finally, we denote by* $R \overset{R}{\leftarrow} \Psi(O)$ *the process of randomly choosing* R *given a fixed* O*, according to the random process* $\Psi^n(O)$ *described above for each* $n \in \mathbb{N}$.

3.2 The Oracle O Provides Adaptive Security

We state, without proof, the following lemma that states that most of the oracles O provide a natural, adaptively secure permutation generator.

Lemma 1. *For all probabilistic, polynomial-time, adaptive adversaries* A *and for all sufficiently* n*:*

$$\Pr_{O \overset{O}{\leftarrow} \Pi}\left[\left|\Pr_{\substack{f \in_{\mathcal{U}} O^n \\ r \in_{\mathcal{U}} \{0,1\}^*}}\left[A^{f,O}(1^n) = 1\right] - \Pr_{\substack{g \in_{\mathcal{U}} \Pi^n \\ r \in_{\mathcal{U}} \{0,1\}^*}}\left[A^{g,O}(1^n) = 1\right]\right| \le \frac{1}{2^{n/2}}\right] \ge 1 - \frac{1}{2^{n/2}},$$

where $r \in_{\mathcal{U}} \{0,1\}^*$ *represents the random coin-tosses of* A.

3.3 The Oracle R Breaks Adaptive Security

Lemma 2. *There exists an efficient adversary, Adv, such that for all oracle pairs* (O, R) *that could possibly be constructed, Adv breaks the adaptive security of* F *relative to* O *and* R.

Proof. We show that the following adversary has a significant chance of distinguishing between the composition of $m(n)$ functions from O and a random function. Note that this adversary calls f *adaptively.*

$Adv^{f,O,R}(1^n)$
 $\boldsymbol{x}_1 = (x_1, \ldots, x_{\ell(n)}) \leftarrow R_1(1^n)$.
 $\boldsymbol{y}_1 = (y_1, ..., y_{\ell(n)}) \leftarrow f(\boldsymbol{x}_1)$.
 $\boldsymbol{x}_2 = (x_{\ell(n)+1}, x_{\ell(n)+2}) \leftarrow R_2(1^n, \boldsymbol{x}_1, \boldsymbol{y}_1)$.
 $\boldsymbol{y}_2 = (y_{\ell(n)+1}, y_{\ell(n)+2}) \leftarrow f(\boldsymbol{x}_2)$.
 If $\perp = R_3(1^n, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}_1, \boldsymbol{y}_2)$ output 0.
 Otherwise output 1.

Fix the oracles O and R. We show that if f was chosen from F then we output 1 and otherwise (w.h.p.) we output 0. It is an easy observation that if $f \in F$ then, by the construction of Adv and R, the adversary *necessarily* outputs 1. Alternatively, if $f \in \diamond^n$ then it is easy to see by the following claim that there is not likely to be any key $\boldsymbol{k}$ where $O_{\boldsymbol{k}}(\boldsymbol{x}) = \boldsymbol{y}$ holds, and therefore the oracle R will output $\perp$, and thus (w.h.p.) Adv will output 0. We remind the reader of the notation defined in Notn. 2, as it is used in the statement of the claim.

Claim 1. For all sufficiently large n, for $\boldsymbol{x} = (x_1, \ldots, x_{\ell(n)}) \leftarrow R(1^n)$:

$$\Pr[f \in_{\mathcal{U}} \Pi^n \;::\; \exists \boldsymbol{K} \in \{0,1\}^{n \cdot m(n)} \; s.t. \; f(\boldsymbol{x}) = O_{\boldsymbol{K}}(\boldsymbol{x})] \leq 2^{-n}.$$

Proof. Let $S = \{O_{\boldsymbol{K}}(\boldsymbol{x}) | \boldsymbol{K} \in \{0,1\}^{n \cdot m(n)}\}$. Clearly $|S| \leq 2^{n \cdot m(n)}$. Consider the probability that $f(\boldsymbol{x}) \in S$, and since $f \in_{\mathcal{U}} \Pi^n$ it is easy to see that this probability is bound by $2^{n \cdot m(n)} / \prod_{i=1}^{\ell(n)} (2^n - i) \leq 2^{n \cdot (m(n) - \ell(n) - 1)} < 2^{-n}$, as $\ell(n) = m(n) + 1$. $\square$

3.4 Simulating the Oracle R for Non-Adaptive Adversaries

It needs to be shown that R does not destroy the non-adaptive security of O. We show that for every non-adaptive adversary with access to the oracle R we can construct another non-adaptive adversary that is essentially just as successful at breaking O, but that has no access to R. Since O is a large set of random permutations, it is clear that without R there can be no successful distinguishing adversary, and therefore there must be no successful non-adaptive adversary relative to R either.

We will begin by showing that for every adversary B relative to R, there exists an adversary $\widehat{B}$ that distinguishes nearly as well as B, but does not make queries to R_3. This is done by having $\widehat{B}$ simulate the responses of R_3. In this simulation there are two general cases: first, there are queries which are likely to be made, and in these cases it turns out that $\widehat{B}$ can simulate R_3's responses with only access to O. Next, there are queries that are unlikely to be made, and we cannot simulate R_3's responses in these cases: we show it is incredibly unlikely that B will make such queries, and thus incorrect answers will not significantly affect the acceptance probability of $\widehat{B}$. Finally, it is then a simple observation that $\widehat{B}$ can easily simulate R_1 and R_2 perfectly, and thus there is no need for $\widehat{B}$ to query the oracle R.

In order to construct $\widehat{B}$ we need B to be in a normal form. First, we assume that an adversary never makes the same oracle query twice. Any adversary that does can be converted to one that does not by storing all of its previous oracle queries and the corresponding responses; it can then look up responses on duplicate queries. Next, for our adversary B, with access to a function oracle $f : \{0,1\}^n \to \{0,1\}^n$, we assume without loss of generality that $B^{f,O,R}(1^n)$ always makes exactly $T(n)$ combined queries to f, O and R, for some polynomial T. Further, we will assume that B records all of its queries and their corresponding responses on its tape in a manner which is efficiently retrievable. In particular, we will assume that for each $k \in \{0,1\}^n$ there is a set $Q_k = \{(q,r)\}$ that contains all of the query/response pairs $O_k(q) \to r$ that have been made; a set $Q_f = \{(q,r)\}$ that contains all the query/response pairs to the challenge function, where $f(q) \to r$; and a set SR_2 that contains all the query/response pairs to R_2.

Lemma 3. *Let T be a polynomial. For every oracle adversary B that on input of size n makes $T(n)$ queries, there exists an oracle adversary $\widehat{B}$: $\widehat{B}$ on input of size n makes at most $T(n) \cdot m(n)$ oracle queries; $\widehat{B}$ never queries R_3; and for all sufficiently large n it is the case that*

$$\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O^n :: \widehat{B}^{O,R,f}(1^n) \neq B^{O,R,f}(1^n)] \leq 5 \cdot T(n)/2^{n/2}$$

and

$$\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} \Pi^n :: \widehat{B}^{O,R,f}(1^n) \neq B^{O,R,f}(1^n)] \leq 5 \cdot T(n)/2^{n/2}.$$

Proof. Fix n. We now construct an adversary $\widehat{B}$ that doesn't make queries to R_3. We note that in the statement of the lemma and its proof we don't concern ourselves with the random coin-tosses of B or $\widehat{B}$. It will be obvious from the proof, that the random coin-tosses do not affect any of the probabilities we discuss, and that we could simply fix the random coin tosses of B and prove the theorem for each such sequence.

We will consider a series of hybrid adversaries. Let $\widehat{A}_i(1^n)$ be an adversary that runs $B(1^n)$ but on the first i oracle queries rather than make an oracle query q it runs the sub-routine $G(q)$ and takes the output of $G(q)$ as the result of the query. Before giving the description of G we remind the reader of the notation defined in Notn. 1 and 2. The sub-routine G is defined below:

$G(q)$

If q is not a query to R_3 perform query q and let a be the oracle's response: output a

Otherwise $q = R_3(x_1, ..., x_{l(n)+2}, y_1, ..., y_{\ell(n)+2})$.
$\boldsymbol{x}_1 = (x_1, ..., x_{l(n)})$ and let $\boldsymbol{x}_2 = (x_{l(n)+1}, x_{l(n)+2})$.
$\boldsymbol{y}_1 = (y_1, ..., y_{l(n)})$ and let $\boldsymbol{y}_2 = (y_{l(n)+1}, y_{l(n)+2})$.

(6) If $((\boldsymbol{x}_1, \boldsymbol{y}_1), \boldsymbol{x}_2) \notin SR_2$ output $\perp$.
(7) $\mathcal{K} = \{\boldsymbol{K} \in (\{0,1\}^n \cup \{f\})^{m(n)} \,|((\boldsymbol{x}_1, \boldsymbol{x}_2), (\boldsymbol{y}_1, \boldsymbol{y}_2)) \in Q_{\boldsymbol{K}}\}$.
(8) If $|\mathcal{K}| \neq 1$ output $\perp$.
(9) $\mathcal{K} = \{\boldsymbol{k}\}$.
(10) If $f \in \boldsymbol{k}$ output $\perp$
(11) Choose $x^* \in_{\mathcal{U}} \{0,1\}^n$ and query $y^* \leftarrow O_{\boldsymbol{k}}(x^*)$.
(12) Output (x^*, y^*).

The intuition behind G is the following: for any query q to an oracle other than R_3 it behaves *identically* to that oracle on query q; for queries to R_3 it will almost surely output the same result as the query to R_3. We give a quick outline of the intuition for the latter case. First, in Line 6, when G outputs $\perp$ it is almost surely the correct answer because if $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ has not been queried from R_2, then the probability of the adversary guessing $\boldsymbol{x}_2$ correctly is negligible. On Line 8 we really have two cases: first, when $|\mathcal{K}| = 0$, it is unlikely there is a key, $\boldsymbol{K}$, such that $O_{\boldsymbol{K}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)$, and thus the response $\perp$ is almost surely correct; next, when $|\mathcal{K}| \geq 2$, and in this case G will output the incorrect answer, but the probability of this case occurring is negligible. The intuition for Line 10 is really the main point behind the proof. If the adversary manages to find a key where it can substitute the function f for part of the key, then the simulation G will output an incorrect answer. However, because of the iterative nature in which $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are exposed and the adversary's limitation of accessing f in a non-adaptive manner, we show that the probability of this event occurring is negligible. Finally, If (x^*, y^*) is output on Line 12 then the output is almost surely correct.

We now look at the cumulative errors that can be made in the hybrid process. We use the following two lemmas that are proven in Section 4.

Lemma 4. *For all sufficiently large n:*

$$\Pr[O \overset{O}{\leftarrow} \Pi; R \overset{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O^n :: \widehat{A}_{i+1}^{O,R,f}(1^n) \neq \widehat{A}_i^{O,R,f}(1^n)] \leq 5/2^{n/2}$$

Lemma 5. *For all sufficiently large n:*

$$\Pr[O \overset{O}{\leftarrow} \Pi; R \overset{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} \Pi^n :: \widehat{A}_{i+1}^{O,R,f}(1^n) \neq \widehat{A}_i^{O,R,f}(1^n)] \leq 5/2^{n/2}$$

We note that by the previous two lemmas, the probabilities that B and $\widehat{A}_{T(n)+1}$ have differing outputs in the same experiments is less than $T(n) \cdot 5/2^{n/2}$, and, since T is a polynomial, this is a negligible amount. Let $\widehat{B}$ be the Turing machine $\widehat{A}_{T(n)+1}$. We note that by inspection of G, and remembering that the call to $O_{\boldsymbol{k}}(x^*)$ on Line 11 of G can mask $m(n)$ queries, $\widehat{B}(1^n)$ makes $\widehat{T}(n) \cdot m(n)$ queries. Further, the probability that $B^{O,R,f}(1^n)$ and $\widehat{B}^{O,R,f}(1^n)$ have differing outputs for the either experiment defined in Lemma 3 is less than $T(n) \cdot 5/2^{n/2}$. $\square$

The last remaining step is to get rid of the queries to R_1 and R_2 that are made by $\widehat{B}$. We note that the results of queries to R_1 and R_2 are independent of O and the challenge function f, and since the results of such queries are random bit strings, they are easy to simulate. Specifically, we consider a Turing machine C that executes $\widehat{B}$ faithfully, but before beginning the simulation $C(1^n)$ will randomly select $(x_1, ..., x_{\ell(n)}) \in_{\mathcal{U}} (\{0,1\}^n)^{[\ell(n)]}$. During the simulation of $\widehat{B}(1^n)$, if there is a query to $R_1(1^n)$, it will respond with $(x_1, ..., x_{\ell(n)})$ and if for $y_1, \ldots, y_{\ell(n)} \in \{0,1\}^n$ there is a query to $R_2(1^n, x_1, ..., x_\ell(n), y_1, .., y_\ell(n))$ it responds with $(x'_{\ell(n)+1}, x'_{\ell(n)+2}) \in_{\mathcal{U}} \left(\{0,1\}^n \setminus \{x_1, \ldots, x_{\ell(n)}\}\right)^{[2]}$. Note that this simulation is perfect. We can now prove the final result of this section.

Lemma 6. *For every probabilistic, polynomial-time, non-adaptive adversary A and for all sufficiently large n:*

$$\Pr_{\substack{O \stackrel{\Omega}{\leftarrow} \Pi \\ R \stackrel{R}{\leftarrow} \Psi(O)}} \left[\left| \Pr_{f \in_{\mathcal{U}} O^n} \left[A^{O,R,f}(1^n) = 1 \right] - \Pr_{g \in_{\mathcal{U}} \Pi^n} \left[A^{O,R,g}(1^n) = 1 \right] \right| \leq \frac{1}{2^{n/3}} \right] \geq 1 - \frac{1}{2^{n/2}},$$

where the $r \in \{0,1\}^*$ represent the random coin-tosses made by A.

Proof. Assume for contradiction that there exists a probabilistic, polynomial time adversary B and infinitely many n for which:

$$\Pr_{\substack{O \stackrel{\Omega}{\leftarrow} \Pi \\ R \stackrel{R}{\leftarrow} \Psi(O)}} \left[\left| \Pr_{\substack{f \in_{\mathcal{U}} O^n \\ r \in \{0,1\}^*}} \left[B^{O,R,f}(1^n) = 1 \right] - \Pr_{\substack{g \in_{\mathcal{U}} \Pi^n \\ r \in \{0,1\}^*}} \left[B^{O,R,g}(1^n) = 1 \right] \right| > \frac{1}{2^{n/3}} \right] > \frac{1}{2^{n/2}}$$

By Lemmas 4 and 5 and the discussion following them, there exists a probabilistic, polynomial time, non-adaptive adversary C that does not query oracle R and infinitely many n such that:

$$\Pr_{\substack{O \stackrel{\Omega}{\leftarrow} \Pi \\ R \stackrel{R}{\leftarrow} \Psi(O)}} \left[\left| \Pr_{\substack{f \in_{\mathcal{U}} O^n \\ r \in \{0,1\}^*}} \left[C^{O,f}(1^n) = 1 \right] - \Pr_{\substack{g \in_{\mathcal{U}} \Pi^n \\ r \in \{0,1\}^*}} \left[C^{O,g}(1^n) = 1 \right] \right| > \frac{1}{2^{n/3-1}} \right] > \frac{1}{2^{n/2}}$$

Observing that the choices over $R \stackrel{R}{\leftarrow} \Psi(O)$ have no effect and can be removed, this result contradicts Lemma 1. □

By using standard counting arguments and the previous lemma, we get the following theorem.

Theorem 2. *There exists a pair of oracles (O, R) where O is a non-adaptively secure permutation generator and where F is not an adaptively secure permutation generator.*

4 Combinatorial Lemmas

4.1 Unique Paths Lemma

An essential point in proving Lemmas 4 & 5 is the following: unless an adversary has already determined by oracle queries to O that for a given key, $\boldsymbol{K}$, of F and ℓ-tuples, $\boldsymbol{x}$ and $\boldsymbol{y}$, where $O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y}$; then the probability that $O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y}$ holds is negligible. The following lemma and its corollary formalizes this concept.

Lemma 7 (Unique Paths Lemma). *Let T, ℓ and m be polynomials. For all sufficiently large $n \in \mathbb{N}$: let $\boldsymbol{x} = (x_1, ..., x_{\ell(n)}), \boldsymbol{y} = (y_1, ..., y_{\ell(n)}) \in (\{0,1\}^n)^{[\ell(n)]}$; for each $i \in \{0,1\}^n$ there is a set $Q_i \subseteq (\{0,1\}^n)^2$ such that $\sum_{i \in \{0,1\}^n} |Q_i| \leq$*

$T(n)$; let $\boldsymbol{K} = (k_1, .., k_{m(n)}) \in (\{0,1\}^n)^{m(n)}$ such that there is no i where $(x_i, y_i) \in Q_{\boldsymbol{K}}$, then:*

$$\Pr[O \stackrel{O}{\leftarrow} \Pi :: O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y} | \forall i,\ \forall (a,b) \in Q_i,\ O_i(a) = b] \leq 2^{(n-1)\cdot \ell(n))}.$$

Proof. We consider several cases. First, we consider the case that there exists a pair $(a, b) \in Q_{\boldsymbol{K}}$ such that either there exists an i s.t. $x_i = a$ but $y_i \neq b$ or there exists a j s.t. $y_j = b$ but $x_j \neq a$. In this case it is not possible for $O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y}$, so the probability is 0.

Second, we consider the case that there exists a $k_i \in \boldsymbol{K}$ where $Q_{k_i} = \{\}$. A necessary condition for $O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y}$ is that $O_{k_i}\left(O_{k_{i-1},\ldots,k_1}(\boldsymbol{x})\right) = O^{-1}_{k_{i+1},\ldots,k_{m(n)}}(\boldsymbol{y})$. The probability of this event is no more than $\prod_{j=1}^{\ell(n)}(1/(2^n - i)) \leq \frac{1}{2^{(n-1)\cdot\ell(n)}}$ (for sufficiently large n).

Thirdly, we consider the case where for every $k_i \in \boldsymbol{K}$ the corresponding set Q_{k_i} is not empty. Because of our conditioning on the probability, for each x_i there exist a value k_j where $\alpha_i = O_{k_1,\ldots,k_{j-1}}(x_i)$ and $\beta_i = O^{-1}_{k_{m(n)},\ldots,k_{j+1}}(y_i)$, but $(\alpha_i, \beta_i) \notin Q_{k_j}$, as otherwise $(x_i, y_i) \in Q_{\boldsymbol{K}}$ which is not permitted by the statement of the lemma. Therefore, the probability that $O_{k_j}(\alpha_i) = \beta_i$ is less than $\frac{1}{2^n - |Q_{k_j}| - \ell(n)}$ (We subtract $\ell(n)$ in the denominator as several x_i's may have this condition occur for the same key k_j.). Therefore, the probability that $O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y}$ is less than $\prod_{i=1}^{\ell(n)} \frac{1}{2^n - |Q_i| - \ell(n)} \leq \frac{1}{2^{(n-1)\cdot\ell(n)}}$, for sufficiently large n (remembering $|Q_i| \leq T(n)$). □

Corollary 1. *Let T, ℓ and m be polynomials. For all sufficiently large $n \in \mathbb{N}$: let $\boldsymbol{x} = (x_1, ..., x_{\ell(n)}), \boldsymbol{y} = (y_1, ..., y_{\ell(n)}) \in (\{0,1\}^n)^{[\ell(n)]}$; for each $i \in \{0,1\}^n$ there is a a set $Q_i \subseteq (\{0,1\}^n)^2$ such that $\sum_{i\in\{0,1\}^n} |Q_i| \leq T(n)$; let $KS \subseteq (\{0,1\}^n)^{m(n)}$ such that for each $\boldsymbol{K} \in KS$ there is no i such that $(x_i, y_i) \in Q_{\boldsymbol{K}}$, then:*

$$\Pr[O \stackrel{O}{\leftarrow} \Pi :: \exists \boldsymbol{K} \in KS \text{ s.t. } O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y} | \forall i,\ \forall (a,b) \in Q_i,\ O_i(a) = b] \leq 2^{n\cdot(m(n)-\ell(n))-\ell(n)}.$$

Proof. This proof follows directly from Lemma 7 and a union bound over the probabilities of each of the keys $\boldsymbol{K} \in KS$. □

4.2 Proof of Lemma 4

For the convenience of the reader, we restate Lemma 4.

Lemma 8. *For all sufficiently large n:*

$$\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O^n :: \widehat{A}^{O,R,f}_{i+1}(1^n) \neq \widehat{A}^{O,R,f}_{i}(1^n)] \leq \frac{5}{2^{n/2}}$$

Proof. We note that in the statement of the lemma and its proof we don't concern ourselves with the random coin-tosses of A_i or A_{i+1}. It will be obvious from the proof, that the random coin-tosses do not affect any of the probabilities we discuss, and that we could simply fix the random coin tosses of B and prove the theorem for each such sequence.

We begin by proving upper-bounds on two probabilistic events that we will frequently want to condition on. We will frequently want to bound the probability that A_{i+1} makes any of its first i queries to O_k, where $f = O_k$. We will call such an event $\mathbf{F}$.

Claim 2. For all sufficiently large n: $\Pr[O \overset{O}{\leftarrow} \Pi; R \overset{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O^n :: \widehat{A}_{i+1}^{f,O,R}(1^n)$ *makes a query to* $O_k = f$ *in one of the first* i *calls to* $G] \leq 2i/2^n$.

Proof. Observe that queries to R_1 and R_2 are statistically independent of f and O. Further, the first i queries are all made by G, and therefore there have been no queries to R_3. Thus the probability of making a query to O_k corresponding to f is no more than the probability of drawing at random the unique red ball from a vase of 2^n balls in i draws without replacement, as there are most i different keys on which O can be queried. Therefore, the probability is bound by $\sum_{j=0}^{i-1} 1/(2^n - j) < 2i/2^n$, for sufficiently large n. □

We also frequently want to bound the probability that by $\widehat{A}_{i+1}$'s ith call to G two oracle queries to O (or O and f) have been made that have the same output. We call such queries *collisions* and we denote such an event by $\mathbf{E}$.

Claim 3. For all sufficiently large n: $\Pr[O \overset{O}{\leftarrow} \Pi; R \overset{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O^n$ :: *after* $\widehat{A}_{i+1}^{f,O,R}(1^n)$ *makes* i *calls to* G *there exists* $k \neq j \in \{0,1\}^n \cup \{f\}$ *s.t.* $(a,b) \in Q_i \wedge (c,b) \in Q_j] \leq 2(i \cdot m(n))^2/2^n$.

Proof. We note that since we are only concerned with queries made in the first i calls to G, there have been no queries to R_3. Next, we condition on $\overline{\mathbf{F}}$ from Claim 2, so query results on f and O_k for $k \in \{0,1\}^n$ are independent of each other. It can easily be observed that to maximize the probability of a collision the adversary should make all of its queries to different functions. The structure of G does not necessarily permit this, but this permits an easy upper-bound on the probability of a collision. Since each call to G makes at most $m(n)$ queries, the probability of $\mathbf{E}$ can be upper-bounded by $\sum_{j=1}^{i \cdot m(n)} \frac{j}{2^n} \leq (i \cdot m(n))^2/2^n$ Since for sufficiently large n the probability of event $\mathbf{F}$ is bound by $2i/2^n$, we can bound the probability of the claim by $2(i \cdot m(n))^2/2^n$. □

To prove Lemma 4, we note that any difference in executions between $A_{i+1}^{O,R,f}(1^n)$ and $A_i^{O,R,f}(1^n)$ must occur in G. We will consider the places where G could have an output different from that of the actual query to the oracle, and bound this probability. We note that this can only occur on lines 6, 8, 10 and 12, and we bound the probability of error on each of these lines with the following series of claims. In order to prove the lemma we take the union bound of the errors from these claims.

Claim 4. The probability that G *gives incorrect output on line 6 is less than* $\frac{1}{2^{2n-1}}$ *for all sufficiently large* n.

Proof. The response to query $q = R_3(\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{x}_2, \boldsymbol{y_2})$ will always be $\perp$ unless $R_2(\boldsymbol{x}_1, \boldsymbol{y}_1) = \boldsymbol{x}_2$. If R_2 has not yet been queried, then it is easily seen, by the definition of R_2, that the probability of the adversary correctly guessing $\boldsymbol{x}_2$ in its query to R_3 is $\frac{1}{(2^n-\ell(n))(2^n-\ell(n)-1)}$. For sufficiently large n, this value is upper-bounded by $\frac{1}{2^{2n-1}}$. $\square$

Claim 5. The probability that G gives incorrect output on line 8 is less than $\frac{1}{2^{n/2}}$ *for all sufficiently large n.*

Proof. For this claim, we consider two separate cases: first we consider the case in which $|\mathcal{K}| = 0$ and next we consider the case where $|\mathcal{K}| \geq 2$.
[**Case** $|\mathcal{K}| = 0$]**:** We first show that for query $R_3(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}_1, \boldsymbol{y}_2)$, if we let $\overline{KS} = \{(k_1, ..., k_{m(n)}) | \exists i s.t. (x_i, y_i) \in Q_{k_1,...,k_{m(n)}}\}$, then with high probability $|\overline{KS}| \leq \ell(n)+2$. Next, we show that for each element $\boldsymbol{K} \in \overline{KS}$ that there is a very small chance that $O_{\boldsymbol{K}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)$. We then show that for $\boldsymbol{K} \in (\{0,1\}^n)^{m(n)} \setminus \overline{KS}$ that the chances that $O_{\boldsymbol{K}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)$ holds is very small using the Unique Paths Corollary (Corollary 1).

In order to bound (w.h.p.) the size of $\overline{KS}$ we will condition on event $\overline{\mathbf{E}}$ from Claim 3 (i.e. there are no collisions). Observe that if $\overline{\mathbf{E}}$ holds, then it is not possible for $|\overline{KS}| > \ell(n)+2$, as otherwise by the pigeonhole principle there would be two keys, $\boldsymbol{\kappa} = (\kappa_1, ..., \kappa_{m(n)})$ and $\boldsymbol{\kappa'} = (\kappa'_1, ..., \kappa'_{m(n)})$, where for some a ($1 \leq a \leq \ell(n)+2$) we would have $y_a = O_{\boldsymbol{\kappa}}(x_a) = O_{\boldsymbol{\kappa'}}(x_a)$, and letting j be the largest index where $\kappa_j \neq \kappa'_j$ this implies $O_{\kappa_j}\left(O_{\kappa_1,..,\kappa_{j-1}}(x_a)\right) = O_{\kappa'_j}\left(O_{\kappa'_1,..,\kappa'_{j-1}}(x_a)\right)$, which is a collision and thus this contradicts our conditioning on event $\overline{\mathbf{E}}$.

Next, we condition on $\overline{\mathbf{F}}$ from Claim 2 to ensure that responses for queries to O are statistically independent of responses for queries to f. We now bound the probability that for any specific key $\boldsymbol{K} \in \overline{KS}$ that $O_{\boldsymbol{K}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)$. We wish to consider the probability that for a key $\boldsymbol{K} = (k_1, ..., k_{m(n)}) \in \overline{KS}$ that $(\boldsymbol{y}_1, \boldsymbol{y}_2) = O_{\boldsymbol{K}}(\boldsymbol{x}_1, \boldsymbol{x}_2)$. For each such key $\boldsymbol{K}$ there exists an i s.t. $(x_i, y_i) \notin Q_{\boldsymbol{K}}$ (otherwise $|\mathcal{K}| \geq 1$ contradicting the case we are in) Consider the smallest j such that there exists a $b \in \{0,1\}^n$ where $(x_i, b) \in Q_{k_1...k_{j-1}}$, and such that for every $b' \in \{0,1\}^n$ it is the case that $(x_i, b') \notin Q_{k_1...k_j}$. The probability that $O_{k_j}(b) = O^{-1}_{k_{j+1},...,k_{m(n)}}(y_i)$ is less than $1/(2^n - |Q_{k_j}|) \leq 1/(2^n - i \cdot m(n))$, as at most $i \cdot m(n)$ queries have been made. Therefore, the probability there exists a key $\boldsymbol{K} \in \overline{KS}$ such that $O_{\boldsymbol{K}}(\boldsymbol{x}) = \boldsymbol{y}$ is less than $\frac{\ell(n)+2}{(2^n - i \cdot m(n))}$, as $|\overline{KS}| \leq \ell(n)+2$ by our conditioning on $\overline{\mathbf{E}}$.

For the remaining set of keys $KS = (\{0,1\}^n)^{m(n)} \setminus \overline{KS}$ the Unique Paths Corollary shows that the probability that there exists a key $\boldsymbol{K} \in KS$ such that $O_{\boldsymbol{K}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)$ is no more than $2^{n \cdot (m(n) - \ell(n)) - \ell(n)}$

Therefore, the probability of the case when $|\mathcal{K}| = 0$ is bounded by $\frac{2(i \cdot m(n))^2}{2^n} + \frac{2i}{2^n} + \frac{(\ell(n)+2)}{(2^n - i \cdot m(n))} + 2^{n \cdot (m(n) - \ell(n)) - \ell(n)} < \frac{1}{2^{n/2}}$ (for sufficiently large n), where the first two summands bound the probabilities of events $\mathbf{E}$ and $\mathbf{F}$ respectively.
[**Case** $|\mathcal{K}| \geq 2$]**:** We observe that in order for $|\mathcal{K}| \geq 2$ to occur, the event $\mathbf{E}$ of Claim 3 must occur at least $\ell(n)+2$ times: there must be at least one collision

for each $y \in (\boldsymbol{y}_1, \boldsymbol{y}_2)$ in order for there to be two keys $\boldsymbol{K}_1, \boldsymbol{K}_2 \in \mathcal{K}$ such that $(\boldsymbol{y}_1, \boldsymbol{y}_2) = O_{\boldsymbol{K}_1}(\boldsymbol{x}_1, \boldsymbol{x}_2) = O_{\boldsymbol{K}_2}(\boldsymbol{x}_1, \boldsymbol{x}_2)$. Therefore, we can use the bound on the probability of $\mathbf{E}$ to bound the probability of incorrect output by $\frac{2(i \cdot m(n))^2}{2^n} < \frac{1}{2^{n/2}}$ (for sufficiently large n). $\square$

Claim 6. The probability that G gives incorrect output on line 10 is less than $\frac{1}{2^{n/2}}$.

Proof. We begin by conditioning on $\overline{\mathbf{F}}$, so that responses from queries to O_k, for $k \in \{0,1\}^n$, are independent of the responses of queries from f. We consider two exclusive cases: first, when $\widehat{A}_{i+1}$ queries f before it queries $R_2(\boldsymbol{x}_1, \boldsymbol{y}_1)$; and second, when $\widehat{A}_{i+1}$ queries $R_2(\boldsymbol{x}_1, \boldsymbol{y}_1)$ before it queries f.

[Case that f was queried before $R_2(\boldsymbol{x}_1, \boldsymbol{y}_1)$]: The intuition behind this case is that the adversary needs to construct a key $\kappa = (k_1, .., k_{u-1}, f, k_{u+1}, .., k_{m(n)})$, and perform queries such that $(x_{\ell(n)+1}, y_{\ell(n)+1}), (x_{\ell(n)+2}, y_{\ell(n)+2}) \in Q_\kappa$. We will argue that it is very unlikely that the queries $x_{\ell(n)+1}$ or $x_{\ell(n)+2}$ were made to O_k for any $k \in \{0,1\}^n$ or f before the query $R_2(\boldsymbol{x}_1, \boldsymbol{y}_1)$. Assuming this to be true, a necessary condition to find a κ satisfying our requirements is to make queries $O_k(\alpha) = \beta$, for $\alpha, k \in \{0,1\}^n$, for which there exists a $j, \gamma \in \{0,1\}^n$ such that there was a $(\beta, \gamma) \in Q_j$ at the time of the query to R_2. We show this is unlikely as well.

We begin by bounding the probability that there had been a query of the form $x_{\ell(n)+1}$ or $x_{\ell(n)+2}$ before the query to R_2, Assume the query $R_2(\boldsymbol{x}_1, \boldsymbol{y}_1)$ was the jth query ($j \leq i$), then the probability that there exists a $\beta, k \in \{0,1\}^n$ such that $(x_{\ell(n)+1}, \beta) \in Q_k$,$(x_{\ell(n)+2}, \beta) \in Q_k$,$(x_{\ell(n)+1}, \beta) \in Q_f$ or $(x_{\ell(n)+2}, \beta) \in Q_f$ is less than $\frac{2 \cdot j \cdot m(n)}{2^n - \ell(n) - 2}$. Next, we condition on that event not happening, and show that there is a small probability that any of the $(j+1)$st through ith queries are of the form $O_k(a) = b$, where there exists a $c, v \in \{0,1\}^n$ such that $(b, c) \in Q_v$ or $(b, c) \in Q_f$ is small. This probability can easily be bounded by $\sum_{s=j}^{i+1} \frac{i+1}{2^n - s \cdot m(n)}$ Therefore, the probability of the first case is less than $\frac{2 \cdot j \cdot m(n)}{2^n - \ell(n) - 2} + \sum_{s=j}^{i+1} \frac{(i+1)}{2^n - s \cdot m(n)} \leq \frac{2 \cdot i \cdot m(n)}{2^n - \ell(n) - 2} + \frac{(i+1)^2}{2^n - (i+1) \cdot m(n)} \leq 2^{-2n/3}$.

[Case $R_2(\boldsymbol{x}_1, \boldsymbol{y}_1)$ was queried before f]: In the second case when the adversary queries f it has already queried R_2, and therefore it needs to find a key $\kappa = (\kappa_1, ..\kappa_{u-1}, f, \kappa_{u+1}, ..., \kappa_{m(n)})$ such that for each $t \leq \ell(n)$, $(x_t, y_t) \in Q_\kappa$. A necessary condition is for there to exist an $a \in \{0,1\}^n$ and $y_s \in \boldsymbol{y}_1$ such that $(a, y_s) \in Q_{f, \kappa_{j+1}, \ldots, \kappa_{m(n)}}$. We show the probability of this occurring is small. We begin by showing it is unlikely that after the query to f there will exists an $a, b, c, k \in \{0,1\}^n$ where both $(a, b) \in Q_f$ and $(b, c) \in Q_k$. Likewise, it is unlikely that there will exist an $a \in \{0,1\}^n$ and $y \in \boldsymbol{y}_1$ where $(a, y) \in Q_f$. If neither of these cases hold then, in order to satisfy our necessary condition, a query to O must be made after the query to f in which there exists a $b, k \in \{0,1\}^n$ where $O_k(b) \in \boldsymbol{y}_1$. We show that the probability of this is also low, proving the lemma.

More formally, assume the query to f is the jth query. There can be at most i (parallel) queries made to f. The probability that the queries to f collide with with any previously made queries is less than $\frac{i \cdot j}{2^n - i}$. The probability that the queries to f will output a $y \in \boldsymbol{y}$ is bound by $\frac{i \cdot \ell(n)}{2^n - i}$. Finally, the probability that any queries to O after the query to f will result in $y \in \boldsymbol{y}$ is less than $\frac{m(n) \cdot i}{2^n - (i+1) \cdot m(n)}$. Therefore, by the union bound the probability of the second case can easily be bound by $\frac{3 \cdot i \cdot m(n)}{2^n - (i+1) \cdot m(n)}$.

Therefore, for all sufficiently large n the probability that the entire claim holds is bound by $\frac{3 \cdot i \cdot m(n)}{2^n - (i+1) \cdot m(n)} + 1/2^{2n/3} + 2(i \cdot m(n))^2/2^n \leq 2^{-n/2}$, where the last summand accounts for our conditioning on $\overline{\mathbf{F}}$. □

Claim 7. The probability that G gives incorrect output on line 12 is less than $\frac{1}{2^{n/2}}$.

Proof. The only reason we may have an incorrect output on line 12 is because the output to an actual query to $R_3(1^n, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}_1, \boldsymbol{y}_2)$ is (x^*, y^*) for $\kappa \in_{\mathcal{U}} \{\kappa \in \{0,1\}^{n \cdot m(n)} | O_\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)\}$, $x^* \in_{\mathcal{U}} \{0,1\}^n$ and $y^* = O_\kappa(x^*)$; whereas, G always outputs $O_{\boldsymbol{K}}(x^*)$ for $\boldsymbol{K} \in \mathcal{K}$ and $x^* \in_{\mathcal{U}} \{0,1\}^n$. Thus, even if there exists a $\boldsymbol{K}' \in (\{0,1\}^n)^{m(n)}$, where $\boldsymbol{K} \neq \boldsymbol{K}'$ and $O_{\boldsymbol{K}'}(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)$, there is no possibility for $(x^*, O_{\boldsymbol{K}'}(x^*))$ to be output by G. We show that it is highly unlikely that $\left|\{\kappa \in \{0,1\}^{n \cdot m(n)} | O_\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{y}_1, \boldsymbol{y}_2)\}\right| > 1$, and thus there is rarely an error in output of G on line 12.

The result follows by conditioning on there being no collisions and then applying the Unique Paths Corollary. In particular, assuming $\overline{\mathbf{E}}$ holds then our sets Q satisfy the requirements for the Unique Paths Corollary where $KS = (\{0,1\}^n)^{m(n)} \setminus \mathcal{K}$. Therefore, by the Unique Paths Corollary we can bound the probability by $2^{n \cdot (m(n) - \ell(n)) - \ell(n)}$, and we bound the probability of $\mathbf{E}$ by $2(i \cdot m(n))^2/2^n$. Therefore by the union bound, the probability of error is less than $2^{-n/2}$ for sufficiently large n. □

To finish proving Lemma 4 we simply take the union bound on the probability of errors in Claims 4,5,6 and 7, and this is less than $5/2^{n/2}$ proving the lemma. □

4.3 Proof of Lemma 5

For the convenience of the reader we restate Lemma 5.

Lemma 9. *For all sufficiently large n:*

$$\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} \Pi^n :: \widehat{A}_{i+1}^{O,R,f}(1^n) \neq \widehat{A}_i^{O,R,f}(1^n)] \leq \frac{5}{2^{n/2}}$$

Proof. We note that this proof is basically the same as the proof of Lemma 4 in the previous section. The only portion of the proof of Lemma 4 that relied on the fact that $f \in O$ as opposed to $f \in \Pi$ was Claim 2, which defines the event $\mathbf{F}$ and bound the probability of it occurring; and those claims that conditioned on the event $\overline{\mathbf{F}}$ and then later had to add in a small probability for error in the case that $\mathbf{F}$ held.

We remind the reader that definition of the event $\mathbf{F}$ is that A_{i+1} makes any of its first i queries to O_k, where $f = O_k$. Clearly, in the experiment for Lemma 5 the probability of the event $\mathbf{F}$ is 0, as $f \in \Pi$ and not O. Therefore, the probability of error in this lemma will be smaller than that of Lemma 4. $\square$

5 Other Constructions, Concluding Remarks & Open Questions

The authors note that the basic design of this oracle and the proof techniques of this paper can be naturally lifted to at least one other natural construction: the XOR of functions. The important observation is that the construction needs to have some natural combinatorial property that corresponds to the Unique Paths Lemma, and with XOR such a property exists, although the notion needs a bit of massaging. The authors leave the proof of this claim to a later version of this paper.

The previous observation leads to the question of whether or not there is a simple combinatorial characterization of those constructions that require a non-relativizing proof technique to show they achieve adaptive security. It also leads to a natural quantitative question: what is the lower-bound on the number of calls to a non-adaptively secure function generator in an adaptively secure black-box construction? Recently, there has been some success in getting quantitative lower bounds in such black-box settings [5, 6, 13], and so it is conceivable one could be found in this setting as well.

As mentioned in the introduction, there is currently a known upper-bound of $\Omega(n/\log n)$ calls to a non-adaptive generator in order to achieving black-box adaptive security. Further, the same upper-bound is achieved by two independent constructions. It would be interesting to know whether or not the current constructions are effectively the best possible. A natural question along these lines is whether or not there are any constructions that would give a smaller upper-bound.

6 Acknowledgments

The author would like to thank Charles Rackoff, Omer Reingold, Vladimir Kolesnikov and the anonymous referees for comments and suggestions that substantially improved the presentation of this paper.

References

1. W. Aiello, M. Bellare, G. Di Crescenzo, and R. Vekatesan. Security amplification by composition: The case of doubly-iterated, ideal ciphers. In *Advances in Cryptology - Crypto 98*, pages 390–407, 1998.
2. Theodore Baker, John Gill, and Robert Solovay. Relativizations of the $\mathcal{P} =? \mathcal{NP}$ question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

3. B. Barak, O. Goldreich, S. Goldwasser, and Y. Lindell. Resettably-sound zero-knowledge and its applications. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 116–125. IEEE Computer Society Press, 2001.
4. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the 20th Annual Symposium on Theory of Computing*, pages 103–112. ACM Press, 1988.
5. R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science*, pages 305–313. IEEE Computer Society Press, 2000.
6. Rosario Gennaro, Yael Gertner, and Jonathan Katz. Lower bounds on the efficiency of encryption and digital signature schemes. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 417–425. ACM Press, 2003.
7. Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In IEEE, editor, *41st Annual Symposium on Foundations of Computer Science*, pages 325–335. IEEE Computer Society Press, 2000.
8. Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In IEEE, editor, *42nd IEEE Symposium on Foundations of Computer Science*, pages 126–135. IEEE Computer Society Press, 2001.
9. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
10. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.
11. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. In *Proceedings of the 17th Annual Symposium on Theory of Computing*, pages 291–304. ACM Press, 1985.
12. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 44–61. ACM Press, 1989.
13. Jeong Han Kim, D. R. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *40th Annual Symposium on Foundations of Computer Science*, pages 535–542. IEEE Computer Society Press, 1999.
14. M. Luby and C. Rackoff. Pseudo-random permutation generators and cryptographic composition. In *Proceedings of the 18th Annual Symposium on Theory of Computing*, pages 353–363. ACM, 1986.
15. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17:373–386, 1988.
16. Ueli Maurer and Krzysztof Pietrzak. Composition of random systems: When two weak make one strong. In *The First Theory of Cryptography Conference*, 2004.
17. Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of psuedo-random functions. In *36th Annual Symposium on Foundations of Computer Science*, pages 170–181. IEEE, 1995.
18. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society Press, 1999.
19. D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology – EUROCRYPT 98*, pages 334–345, 1998.