

# 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.

**Key words:** Pseudo-Randomness, Function Generators, Composition, XOR, Adaptive/Non-Adaptive Security, Oracle Separation.

## 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 [15, 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 many in cryptographic community believe this to be true.

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 programme 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 also 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 show how to construct the stronger object from the weaker one. The first is to treat the non-adaptively secure generator as pseudo-random number generator and then use the construction of Goldreich, Goldwasser and Micali [9] in order to construct a pseudo-random function generator.

The second construction is to treat the non-adaptively secure function generator as a synthesizer and then construct 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 [16]. 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 constructions used in practice might be more efficient.

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, almost all constructions in modern cryptography are black-box. More specifically, when given an implementation,  $M_P$ , of a primitive  $P$ , we construct a primitive  $C$  by a construction  $C^{M_P}$ , where  $M_P$  is treated as an oracle. The difference between the two constructions is that in the former case the construction  $C$  may make use of the machine description  $M_P$ , while in the latter it may only treat it as a function to be queried inside of a black-box.

Observe that it is not immediately clear how to prove that there can be no black-box construction  $C^{M_P}$  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  $M_P$ , and independently realize the primitive  $Q$ . In order to address this Impagliazzo and Rudich [12] gave a model in which one can prove separations. In their model they note that black-box constructions and proofs work relative to any oracle, and therefore it is sufficient to provide an oracle  $O$  relative to which implementations  $M_P^O$  of primitive  $P$  exists, but for all constructions  $C^{M_P^O}$  the primitive  $Q$  is not secure relative to  $O$ . Our result will be of this flavour. We note that recently Gertner, Malkin and Reingold have shown that if your goal is to rule out black-box constructions then a weaker type of theorem will suffice. We direct the interested reader to [8].

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  $C$  cannot be proven using black-box techniques by giving oracles  $O$  and implementations of non-adaptively secure function generators relative to them  $M^O$ , for which  $C^{M^O}$  is not adaptively secure.

Finally, we note that there are several non-black box techniques that are used in cryptography, such as Zero-Knowledge in its many incarnations [11, 10, 4, 18, 3](to name a few) and the recent work by Barak [2]. However, the authors feel it is unlikely that these techniques will have application to the problem at hand, as the settings of these works seem dissimilar to the one considered here.

## 1.2 Our Results

**Theorem 1** *Let  $m$  be a polynomial. There exists a pair of oracles  $(O, R)$  relative to which there exist non-adaptively secure pseudo-random permutation generators  $P$ , but where  $\underbrace{P \circ \dots \circ P}_m$  is not*

*adaptively secure.*

In the theorem  $P \circ P'$  denotes the natural composition construction: it is the generator attained by randomly choosing a  $p \in P$  and  $p' \in P'$  and computing the permutation  $p \circ p'$ .

### 1.3 Preliminaries & Notation

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  $\Pr[R_1; \dots; R_k :: E|C]$  to denote the probability that if random processes  $R_1$  through  $R_k$  are performed, in order, that event  $E$  occurs conditioned on event  $C$ . We say a function  $\mu : \mathbb{N} \rightarrow \mathbb{R}$  is *negligible* if for all constants  $c > 0$  and all sufficiently large  $n : \mu(n) \leq \frac{1}{nc}$ .

Finally, let  $D$  be any finite set, and let  $Q_1, Q_2 \subseteq D \times D$  be arbitrary sets of pairs. We define  $Q_1 \circ Q_2 = \{(a, c) | \exists b, b \in D \text{ s.t. } (a, b) \in Q_1 \wedge (b, c) \in Q_2\}$ . For  $\vec{K} = (k_1, \dots, k_m)$ , let  $Q_{\vec{K}} = Q_{k_1} \circ \dots \circ Q_{k_m}$ .

### 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 then show that relative to these oracles that the composition of generators does not necessarily provide adaptive security. Finally, we show that relative to these oracles non-adaptively secure permutation generators do exist. This is done by showing that non-adaptive adversaries cannot make effective use of one of the oracles. This is done by showing how to simulate the responses of said oracle. 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 circuit based, non-uniform definitions for pseudo-random function generators and adversaries. We note that since we are proving a separation, the result is stronger when proved relative to the non-uniform version of the definitions, as opposed to the uniform ones.

We will assume that all of the binary and unary functions are available as gates, as well as the oracle gates discussed later. The size of a circuit  $C$ , denoted  $|C|$ , is the number of edges between gates in the circuit.

**Definition 1 (Function Ensembles)** We call  $G : \{0, 1\}^{\kappa} \times \{0, 1\}^n \rightarrow \{0, 1\}^m$  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  $m$  and  $\ell$  be polynomials, and let  $\mathcal{N} \subseteq \mathbb{N}$  be an infinitely large set. For each  $n \in \mathcal{N}$ , let  $G^n : \{0, 1\}^{\ell(n)} \times \{0, 1\}^n \rightarrow \{0, 1\}^{m(n)}$  be a function generator. We call  $G = \{G^n | n \in \mathcal{N}\}$  a function ensemble. Given an ensemble  $G$ , if for every  $n \in \mathcal{N}$  with probability 1 the function  $g \in G^n$  is a permutation, then we say  $G$  is a permutation ensemble. We say an ensemble  $G$  is efficiently computable if there exists a family of circuits  $\{C_i\}_{i \in \mathbb{N}}$  and a polynomial  $p$  such that for all sufficiently large  $n$  the generator  $G^n$  is computed by  $C_i$  and  $|C_i| < p(n)$ .

**Definition 2 ((Non-)Adaptive Adversaries)** An adversary,  $A = \{A_i\}_{i \in \mathbb{N}}$  is a family of polynomial size circuits with oracle gates. We denote an adversary  $A$  with access to an oracle  $f$  as  $A^f$ . An adversary is adaptive if it can have several oracle gates, where the inputs of one oracle gate  $f$  can depend on the outputs of previous oracle gates.

A non-adaptive adversary can have multiple oracle gates, but there may be no oracle gate whose inputs depend on the outputs of another oracle gate. Intuitively, this corresponds to the adversary making all of its queries to the oracle at the same time.

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

We say that  $G$  is adaptively/non-adaptively secure if for all constants  $c > 0$ , for all adaptive/non-adaptive polynomial sized adversaries  $A = \{A_i\}_{i \in \mathbb{N}}$  and for all sufficiently large  $n$ :

$$\left| \Pr_{g \in \mathcal{U} G^n} [A_n^g = 1] - \Pr_{f \in \mathcal{U} \mathcal{F}^n} [A_n^f = 1] \right| \leq \frac{1}{n^c}.$$

In this work we are concerned with the above definitions, but in worlds where a pair of oracles  $(O, R)$  exist. We extend the definitions of function ensembles and adaptive/non-adaptive adversaries by allowing circuits to have oracle gates for the oracles  $O$  and  $R$ . On an input  $x$  the gates compute the value of the oracle on  $x$ . We stress that non-adaptive adversaries *are* permitted to query  $O$  and  $R$  in a dependent manner: the non-adaptive restriction on oracle queries in the definition of the adversary is only for the oracle  $f$  specified in the definition.

### 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  $p : \{0, 1\}^n \rightarrow \{0, 1\}^n$ . Clearly, a non-adaptively secure PRPG  $F$  can be constructed relative to 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  $\pi_1, \pi_2 \in \mathcal{U} O$ . The oracle  $R$  iteratively requests the values of  $y_i = \pi_1 \circ \pi_2(x_i)$  for enough randomly chosen values  $x_i$  that it should be able to uniquely identify  $\pi_1, \pi_2 \in O$ . If at this point  $R$  has determined a  $\pi_1, \pi_2 \in O$  consistent with all of the  $y$ 's, 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  $O$  that is consistent with the values returned to it then the oracle rejects the input 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  $y$  values  $R$  requests, and if  $R$  returns a prediction  $(X^*, y^*)$  that is consistent with the permutation being distinguished then almost surely the permutation is a composition of permutations from  $O$ . In contrast, if the adversary is *non-adaptive* then the oracle will be of essentially no use to the adversary. This is due to the iterative nature of  $R$ . Therefore, it is as if  $R$  does not exist, and the adversary cannot use it to help distinguish permutations that are in  $O$ .

### 3.1 Oracle Definitions

**Definition 4 (The Oracle  $O$ )** Let  $\Pi^n$  be the set of all permutations  $p : \{0, 1\}^n \rightarrow \{0, 1\}^n$ . Let  $O^n \stackrel{O}{\leftarrow} \Pi^n$  denote the process of choosing an indexed set of  $2^n$  random permutations from  $\Pi^n$  with replacement. Let  $O_k^n$  denote the  $k$ th permutation in  $O^n$ . Let  $O = \{O^n | n \in \mathbb{N}\}$ . Where  $n$  is clear we write  $O_k$  to denote  $O_k^n$ . If  $\vec{K} = (k_1, \dots, k_m) \in \{0, 1\}^{n \times m}$ , then let  $O_{\vec{K}}$  denote  $O_{k_m} \circ \dots \circ O_{k_1}$ . Further, if  $\vec{x} \in \{0, 1\}^{n \times \ell}$  then denote  $O_k(\vec{x}) = \vec{y} = (O_k(x_1), \dots, O_k(x_\ell))$ .

**Definition 5 (Permutation Generator Construction)** The natural construction  $I$  of a permutation generator from an oracle  $O \stackrel{O}{\leftarrow} \Pi$  is to let  $I_k(x) = O_k(x)$ , for all  $k, x \in \{0, 1\}^n$ .

**Definition 6 (Composition Construction)** Let  $I$  be a permutation generator, and let  $m : \mathbb{N} \rightarrow \mathbb{N}$  be a polynomial, so that  $m(i) > 2$  for all  $i \in \mathbb{N}$ . For every  $n \in \mathbb{N}$  and  $\vec{K} \in \{0, 1\}^{m(n) \times n}$  let  $F_{\vec{K}=k_1, \dots, k_{m(n)}}^n(x) = I_{k_1} \circ \dots \circ I_{k_{m(n)}}(x)$ . Let  $F = \cup_n F^n$  be the proposed construction for an adaptively secure PRPG.

**Definition 7 (The Oracle  $R$ )** For an oracle  $O$  as described in Definition 4 and a construction  $F$  of  $m$  compositions as described in Definition 6 we define the oracle as follows. Set with foresight  $\ell(n) = m(n) + 4$  for every  $n \in \mathbb{N}$ . Let  $R = \{(R_1, R_2, R_3)\}$  be an oracle that, for each  $n$ , is chosen randomly and then fixed according to the random process,  $\Psi^n(O)$ , described below:

$R_1(1^n) \rightarrow x_1, \dots, x_{\ell(n)}$  where  $x_i \in_{\mathcal{U}} \{0, 1\}^n$ .

$R_2(1^n, x_1, \dots, x_{\ell(n)}, y_1, \dots, y_{\ell(n)}) \rightarrow x_{\ell(n)+1}, x_{\ell(n)+2}$  where for each  $i$   $y_i \in \{0, 1\}^n$  and  $x_{\ell(n)+1}, x_{\ell(n)+2} \in_{\mathcal{U}} \{0, 1\}^n$ .

$R_3(1^n, x_1, \dots, x_{\ell(n)+2}, y_1, \dots, y_{\ell(n)+2}) = (x^*, y^*)$  where  $\kappa \in_{\mathcal{U}} \{\kappa = (k_1, \dots, k_{m(n)}) \in \{0, 1\}^{n \times m(n)} | O_{k_1, \dots, k_{m(n)}}(x_i) = y_i \forall i, 1 \leq i \leq \ell(n) + 2\}$  and  $x^* \in_{\mathcal{U}} \{0, 1\}^n$  and  $y^* = O_{k_1, \dots, k_{m(n)}}(x^*)$ .

On all other inputs to the oracle  $R^n$  outputs  $\perp$ . Finally, we denote by  $R \stackrel{R}{\leftarrow} \Psi(O)$  the process of randomly choosing  $R$  given a fixed  $O$ , according to the random process described.

### 3.2 The Oracle $R$ Breaks Adaptive Security

**Lemma 1** Relative to  $O$  and  $R$ , the construction  $F$  is not an adaptively secure PRPG.

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.

$Adv^{f, O, R}(1^n)$

$\vec{x}_1 = (x_1, \dots, x_{\ell(n)}) \leftarrow R_1(1^n)$ .

$\vec{y}_1 = (y_1, \dots, y_{\ell(n)}) \leftarrow f(\vec{x}_1)$ .

$\vec{x}_2 = (x_{\ell(n)+1}, x_{\ell(n)+2}) \leftarrow R_2(1^n, \vec{x}_1, \vec{y}_1)$ .

$\vec{y}_2 = (y_{\ell(n)+1}, y_{\ell(n)+2}) \leftarrow f(\vec{x}_2)$ .

If  $\perp = R_3(1^n, \vec{x}_1, \vec{x}_2, \vec{y}_1, \vec{y}_2)$  output 0.

Otherwise output 1.

We show that if  $f$  was chosen from  $F$  then we output 1 and otherwise we output 0 with high probability. 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  $f \notin F$  then it is easy to see by the following claim that there is not likely to be any key  $\vec{k}$  where  $O_{\vec{k}}(\vec{x}) = \vec{y}$  holds, and therefore the oracle  $R$  will output  $\perp$ , and thus  $Adv$  will output 0.

**Claim 1** For all sufficiently large  $n$ , for all  $\vec{x} \in \{0,1\}^{n \cdot m(n)}$ :  $\Pr[O \stackrel{O}{\leftarrow} \Pi; f \in_{\mathcal{U}} \Pi \ :: \ \exists \vec{K} \in \{0,1\}^{m(n) \times n} \text{ s.t. } f(\vec{x}) = O_{\vec{K}}(\vec{x})] \leq 2^{-n}$ .

**Proof:** Let  $S = \{O_{\vec{K}}(\vec{y}) | \vec{K} \in \{0,1\}^{n \cdot m(n)}\}$ . Clearly  $|S| \leq 2^{n \cdot m(n)}$ . Consider the probability that  $f(\vec{x}) \in S$ , and since  $f \in_{\mathcal{U}} \Pi$  it is easy to see that this probability is bound by  $2^{n \cdot m(n)} / \prod_{i=1}^{\ell(n)+2} (2^n - i) \leq 2^{n \cdot (m(n) - \ell(n) + 1)} < 2^{-n}$ , as  $\ell(n) = m(n) + 4$ .

...□

### 3.3 Simulating the Oracle $R$ for Non-Adaptive Adversaries

It needs to be shown that  $R$  does not destroy the non-adaptive security of  $I$ . 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  $I$ , but it has no access to  $R$ . Since  $I$  is just a random permutation 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  $\hat{B}$  that distinguishes nearly as well as  $B$ , but does not make queries to  $R_3$  unless it is likely that there will be a response that is not  $\perp$ . In the case that there is a response that it not  $\perp$ , it turns out it is likely that  $\hat{B}$  could have answered the query with only access to  $O$ . Therefore, the original query was unnecessary. It is then a simple observation that  $\hat{B}$  can easily simulate  $R_1$  and  $R_2$  perfectly, and thus there is no need for  $\hat{B}$  to query the oracle  $R$ .

In order to construct  $\hat{B}$  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 doesn't by storing all of its previous oracle queries and the corresponding responses; it can then look up responses on duplicate queries.

Next, we assume without loss of generality that  $B = \{B_n\}_{n \in \mathbb{N}}$  always make exactly  $T(n)$  queries, for some polynomial  $T$ . We will consider  $B$  to be the composition of a sequence of component circuits  $B = A_1 \circ G \circ A_2 \circ G \circ \dots \circ G \circ A_{T(n)+1}$ , where for every  $i$  the circuit  $A_i$  makes no oracle queries, and the circuit  $G$  in effect only makes oracle queries. More specifically the output of  $A_i$ , ( $1 \leq i \leq T(n)$ ), is a pair  $(\sigma, q)$  where  $\sigma$  is state information and  $q$  is a query to an oracle  $O, f$  or  $R$ . The circuit  $G(\sigma, q)$  outputs  $(\sigma, q, a)$  where  $a$  is the response to the oracle query  $q$ . The input to  $A_i$ , ( $2 \leq i \leq T(n) + 1$ ), is a triple  $(\sigma, q, a)$  where  $a$  is the answer to the oracle query  $q$ . Note  $A_1$  has no input, and  $A_{T(n)+1}$  has as output a single bit  $b$ . Further, we will assume that  $\sigma = (\sigma', \{Q_k | Q_k \neq \emptyset\}, Q_f, SR_2)$ ; where  $Q_k = \{(q, r)\}$  contains all the queries/response pairs  $O_k(q) \rightarrow r$  that have been made;  $Q_f = \{(q, r)\}$  contains all the query/response pairs  $f(q) \rightarrow r$  to  $f$ ;  $SR_2$  contains all the query/response pairs to  $R_2$ ; and  $\sigma'$  contains all other extraneous state information.

**Lemma 2** Let  $T$  be a polynomial. For every adversary  $B = \{B_n\}$ , where  $B_n$  makes  $T(n)$  queries, there exists an adversary  $\hat{B} = \{\hat{B}_n\}$ :  $\hat{B}_n$  makes at most  $\hat{T}(n)$  queries for a polynomial  $\hat{T}$ ;  $\hat{B}$  never queries  $R_3$ ;  $\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O \ :: \ \hat{B}^{\mathcal{O}, R, f} \neq B^{\mathcal{O}, R, f}] \leq \mu(n)$ ; and  $\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} \Pi \ :: \ \hat{B}^{\mathcal{O}, R, f} \neq B^{\mathcal{O}, R, f}] \leq \mu(n)$ , where  $\mu$  is a negligible function.

**Proof:** Fix  $n$ . We now construct an adversary  $\widehat{B}$  that doesn't make queries to  $R_3$ . We will consider a series of hybrid adversaries. Let

$$\widehat{A}_i = A_1 \circ \overline{G} \circ A_2 \circ \dots \circ A_{i-1} \circ \overline{G} \circ A_i \circ G \circ A_{i+1} \circ G \dots \circ G \circ A_{T(n)+1},$$

where  $\overline{G}$  is defined as:

- $$\begin{aligned} &\overline{G}(\sigma = (\sigma', \{Q_i | Q_i \neq \emptyset\}, SR_2), q) \\ &\quad \text{If } q \text{ is not a query to } R_3 \text{ query } q \text{ and let } a \text{ be the oracle's response: output} \\ &\quad (\sigma, q, a) \\ &\quad \text{Otherwise } q = R_3(x_1, \dots, x_{l(n)+2}, y_1, \dots, y_{l(n)+2}). \\ &\quad \text{Let } \vec{x}_1 = (x_1, \dots, x_{l(n)}). \\ &\quad \text{Let } \vec{x}_2 = (x_{l(n)+1}, x_{l(n)+2}). \\ &\quad \text{Let } \vec{y}_1 = (y_1, \dots, y_{l(n)}). \\ &\quad \text{Let } \vec{y}_2 = (y_{l(n)+1}, y_{l(n)+2}). \\ (8) \quad &\text{If } ((\vec{x}_1, \vec{y}_1), \vec{x}_2) \notin SR_2 \text{ output } (\sigma, q, \perp). \\ &\mathcal{K} = \{\vec{k} = (k_1, \dots, k_{m(n)}) \in (\{0, 1\}^n \cup \{f\})^{m(n)} | ((\vec{x}_1, \vec{x}_2), (\vec{y}_1, \vec{y}_2)) \in Q_{\vec{k}}\}. \\ (10) \quad &\text{If } |\mathcal{K}| = 0 \text{ output } (\sigma, q, \perp). \\ (11) \quad &\text{If } |\mathcal{K}| > 1 \text{ output } (\sigma, q, \perp) \\ (12) \quad &\text{Let } \mathcal{K} = \{\vec{k}\}. \\ (13) \quad &\text{If } f \in \vec{k} \text{ output } (\sigma, q, \perp) \\ &\text{Otherwise } (x^*, y^*) \leftarrow R_3(\vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_2). \\ &\text{Query } \hat{y}^* \leftarrow O_{\vec{k}}(x^*). \\ (16) \quad &\text{If } \hat{y}^* \neq y^* \text{ output } (\sigma, q, \hat{y}^*). \\ (17) \quad &\text{Output } (\sigma, q, \hat{y}^*). \end{aligned}$$

The intuition behind  $\overline{G}$  is that it only performs those queries to  $R_3$  that are unlikely to result in an output of  $\perp$  and those queries that are likely to occur in the first place. Thus on lines 8 and 10 when we output  $\perp$  it is because it is most likely the correct answer. On the other hand, the output given on lines 11, 13 and 16 is most likely wrong, but the probability of it occurring is negligible.

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

**Lemma 3** *There exists an explicit negligible function  $\epsilon$  such that for all sufficiently large  $n$ :*

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

**Lemma 4** *There exists an explicit negligible function  $\delta$  such that for all sufficiently large  $n$ :*

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

We note that by the previous two lemmas, the probabilities that  $B$  and  $\widehat{A}_{T(n)+1}$  have differing outputs in the same experiment is less than  $T(n) \cdot \max\{\epsilon(n), \delta(n)\}$ , and since  $T$  is a polynomial and  $\epsilon$  and  $\delta$  are both negligible, this is a negligible amount. Next, observe that in  $\overline{G}$  the call to  $R_3$  on line 14 is being used as a glorified random generator of  $x^*$ , as the value  $y^*$  is necessarily ignored. Therefore, we can replace line 14 with the line:  $x^* \in_{\mathcal{U}} \{0, 1\}^n$ , and  $\overline{G}$  will function exactly as before. Under this modification we see that  $\widehat{A}_{T(n)+1}$  never queries  $R_3$ . Let  $\widehat{B}$  be this circuit. We note that  $\widehat{B}$  makes no more than  $\widehat{T}(n) = T(n) \cdot m(n)$  queries, and that  $\mu(n) = T(n) \cdot \max\{\epsilon(n), \delta(n)\}$ .

...□

The last step remaining 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  $f$ , and since they are basically random numbers, they're easy to simulate. Specifically, we consider a circuit  $C$  that executes  $\widehat{B}$  faithfully, unless there is a query to  $R_1(1^n)$ , in which case it responds with  $x_1, \dots, x_{\ell(n)} \in_{\mathcal{U}} \{0, 1\}^{n \times m(n)}$ , and if there is a query to  $R_2(1^n, x'_1, \dots, x'_{\ell(n)}, y_1, \dots, y_{\ell(n)})$  it responds with  $(x'_{\ell(n)+1}, x'_{\ell(n)+2}) \in_{\mathcal{U}} \{0, 1\}^{n \times 2}$ . Note that this simulation is perfect. We now prove the final result of this section.

**Lemma 5** *Relative to  $O$  and  $R$  the construction  $I$  is non-adaptively secure.*

**Proof:** Assume the opposite for contradiction that there exists an adversary  $B$  and constant  $d > 0$  so for infinitely many  $n$ :

$$\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O; f \in_{\mathcal{U}} \Pi :: B^{O,R,f} = 1 - B^{O,R,f} = 1] \geq \frac{1}{n^d}.$$

By the previous discussion this implies there exists polynomial sized adversary  $C$  such that:

$$\Pr[O \stackrel{O}{\leftarrow} \Pi; f \in_{\mathcal{U}} O; g \in_{\mathcal{U}} \Pi :: C^{O,g} = 1 - C^{O,g} = 1] \geq \frac{1}{n^d} - 2 \cdot \mu(n) \geq \frac{1}{n^{d'}},$$

for some  $d' > 0$  and infinitely many  $n$ , as  $\mu$  is a negligible function. However, this contradicts the provable security of  $O$  when  $R$  is not present, and therefore our assumption about the distinguishing capabilities of  $B$  are incorrect.

...□

## 4 Combinatorial Lemmas

### 4.1 Unique Paths Lemma

An essential point in proving Lemmas 3 & 4 is the following: unless an adversary has determined by oracle queries to  $O$  that for a given  $\vec{K}, \vec{x}$  and  $\vec{y}$  that  $O_{\vec{K}}(\vec{x}) = \vec{y}$ , then the probability that  $O_{\vec{K}}(\vec{x}) = \vec{y}$  is negligible. The following lemma formalizes this concept.

**Lemma 6 (Unique Paths Lemma)** *Let  $q, \kappa, \ell$ , and  $m$  be polynomials. For all sufficiently large  $n \in \mathbb{N}$ , let  $\vec{x} = (x_1, \dots, x_{\ell(n)})$ ,  $\vec{y} = (y_1, \dots, y_{\ell(n)}) \in \{0, 1\}^{n \times \ell(n)}$ ; let  $KS \subseteq \{0, 1\}^{n \times m(n)}$ ; for each  $i \in \{0, 1\}^n$  let  $Q_i = \{(a, b) | O_i(a) = b\} \subseteq \{0, 1\}^{n \times 2}$ ; and let  $Q = \{Q_k | k \in \{0, 1\}^n \wedge 1 \leq |Q_k| \leq q(n)\}$  where  $\forall \vec{K} \in KS, \forall i (x_i, y_i) \notin Q_{\vec{K}}$  and  $|Q| \leq \kappa(n)$ :*

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

**Proof:** We will independently consider the probability of every key  $\vec{K} \in KS$  that  $O_{\vec{K}}(\vec{x}) = \vec{y}$  and take the union bound of these probabilities.

The first case we consider are the keys  $\vec{K} \in KS$  where there exists a pair  $(a, b) \in Q_{\vec{K}}$  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  $x_j \neq a$ . The probability of success for these keys is 0.

Next we consider the case of all  $\vec{K} \in KS$  that contain a key  $k_i \in \vec{K} = (k_1, \dots, k_{m(n)})$  such that  $Q_i \notin Q$ . A necessary condition for  $\vec{y} = O_{\vec{K}}(\vec{x})$  is that  $O_{k_i}(O_{k_{i-1}, \dots, k_1}(\vec{x})) = O_{k_{i+1}, \dots, k_{m(n)}}^{-1}(\vec{y})$ . The

probability of this event is no more than  $\prod_{j=1}^{\ell(n)} (1/(2^n - i)) \leq \frac{1}{2^{(n-1)\ell(n)}}$  (for sufficiently large  $n$ ). The number of sequences that contain some  $k \notin \vec{K}$  is bounded by  $2^{n \cdot m(n)}$ . By the union bound, the probability that any such  $\vec{K}$  that  $O_{\vec{K}}(\vec{x}) = \vec{y}$  is less than  $2^{n \cdot (m(n) - \ell(n) + 1)}$ .

Finally, consider the case of sequences  $\vec{K} = (k_1, \dots, k_{m(n)}) \in KS$  where for each  $k \in \vec{K}$  the set  $Q_k \in Q$ . Fix such a  $\vec{K}$ . Because we are conditioning on  $Q$ , for each  $x_i$  there exist value  $k_j$  where for  $\alpha_i = O_{k_1, \dots, k_{j-1}}(x_i)$  and  $\beta_i = O_{k_{\ell(n)+2}, \dots, k_{j+1}}(y_i)$  that  $(\alpha_i, \beta_i) \notin Q_{k_j}$ , as otherwise  $(x_i, y_i) \in Q_{\vec{K}}$  which is not permitted by the definition of  $Q$ . Therefore, the probability that  $O_{k_j}(\alpha_i) = \beta_i$  is less than  $\frac{1}{2^{n - |Q_{k_j}| - i}}$  (We subtract  $i$  in the denominator, as we don't the number of  $x$ 's for which this condition occurs on  $k_j$ , but we can bound it to being less than  $i$ , as we are currently concerned with  $x_i$ ). Therefore, the probability that  $O_{\vec{K}}(\vec{x}) = \vec{y}$  is less than  $\prod_{i=1}^{\ell(n)} \frac{1}{2^{n - |Q_i| - i}} \leq \frac{1}{2^{(n-1)\ell(n)}}$ , for sufficiently large  $n$  (remembering  $|Q_i| \leq q(n)$ ). For the case under consideration, the number of sequences  $\vec{K} = k_1, \dots, k_{m(n)} \in KS$  that need to be considered is bound by  $\kappa(n)^{m(n)}$ , and therefore the total probability of  $O_{\vec{K}}(\vec{x}) = \vec{y}$  for such sequences is less than  $2^{n \cdot (m(n) - \ell(n) + 1)}$ .

Taking the union bound on all three cases, we see that the probability is smaller than  $2^{n \cdot (m(n) - \ell(n) + 1) + 1} < 2^{n \cdot (m(n) - \ell(n) + 2)}$ .

...□

## 4.2 Proof of Lemma 3

For the convenience of the reader we restate Lemma 3, but in this case we specify  $\epsilon(n) = \frac{5}{2^{n/2}}$ .

**Lemma 7** For all sufficiently large  $n$ :

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

**Proof:** 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 **F**.

**Claim 2**  $\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O :: \widehat{A}_{i+1}$  makes one of its first  $i$  queries to  $O_k$  where  $f = O_k] \leq 2i/2^n$  for sufficiently large  $n$ .

**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  $\overline{G}$ , and therefore the replies to any queries to  $R_3$  are dependent only on the set of keys  $\mathcal{S} = \{k' | \widehat{A}_{i+1} \text{ has queried } O_{k'}\}$ . This means if any of the first  $i$  queries are to  $R_3$  then they do not influence the probability of making a query to  $O_k$ . Therefore, the probability is bound by  $\sum_{j=1}^i 1/(2^n - i) < 2i/2^n$ , as there are at most  $i$  different keys  $k$  that can be queried.

...□

We also frequently want to bound the probability that by its  $i$ th query the adversary  $\widehat{A}_{i+1}$  has made two queries to  $O$  that had the same output. We call such queries collisions and we denote such an event by **E**.

**Claim 3**  $\Pr[O \stackrel{O}{\leftarrow} \Pi; R \stackrel{R}{\leftarrow} \Psi(O); f \in_{\mathcal{U}} O :: \widehat{A}_{i+1}^{\mathcal{O}, R, f}$  after  $i$  queries  $\exists i \neq j$  s.t.  $(a, b) \in Q_i \wedge (c, b) \in Q_j] \leq 2(i \cdot m(n))^2/2^n$  for sufficiently large  $n$ .

**Proof:** We condition on  $\overline{\mathbf{F}}$  from Claim 2, as this way query results on  $f$  and  $O_k$  ( $k \in \{0, 1\}^n$ ) are independent of each other.

We note since we are only concerned with the first  $i$  queries, then all queries made are to  $O, f, R_1$  and  $R_2$ . To observe that queries to  $R_3$  are not involved note that  $\hat{A}_{i+1}$  would have the *exact* same acceptance probability, if  $\overline{G}$  were modified to chose  $x^* \in_{\mathcal{U}} \{0, 1\}^n$  and never called  $R_3$ . Therefore, the probability of  $\mathbf{E}$  can be upper-bounded by  $\sum_{j=1}^{i \cdot m(n)} \frac{1}{2^{n-j}} \leq (i \cdot m(n))^2 / 2^n$  (for sufficiently large  $n$ ). Since  $\mathbf{F}$  occurs with probability less than  $2i/2^n$ , we can bound the probability of the claim by  $2(i \cdot m(n))^2 / 2^n$  (for sufficiently large  $n$ ).

...□

To prove the lemma we note that any difference in executions between  $A_{i+1}$  and  $A_i$  must occur in  $\overline{G}$ . We will consider the places where  $\overline{G}$  could have an output different from  $G$ , and bound this probability. We note that this can only occur on lines 8, 10, 11, 13 and 16, and we bound the probability of error on each of these lines with the following series of claims. We will then take the union bound of the errors.

**Claim 4** *The probability that  $\overline{G}$  gives incorrect output on line 8 is less than  $\frac{1}{2^{2n}}$ .*

**Proof:** The response to query  $q = R_3(\vec{x}_1, \vec{y}_1, \vec{x}_2, \vec{y}_2)$  will always be  $\perp$  unless  $R_2(\vec{x}_1, \vec{y}_1) = \vec{x}_2$ . By the definition of  $R_2$ , the probability of this event is less than  $(1/2)^{2n}$ .

...□

**Claim 5** *The probability that  $\overline{G}$  gives incorrect output on line 10 is less than  $2^{-n/2}$ .*

**Proof:** We first show that for query  $R_3(\vec{x}_1, \vec{x}_2, \vec{y}_1, \vec{y}_2)$ , if we let  $\overline{KS} = \{(k_1, \dots, k_{m(n)}) \mid \exists \text{is.t. } (x_i, y_i) \in Q_{k_1} \circ \dots \circ Q_{k_{m(n)}}\}$ , then with high probability  $|\overline{KS}| \leq \ell(n) + 2$ . Next, we show that for each element  $\vec{K} \in \overline{KS}$  that there is a very small chance that  $O_{\vec{K}}(\vec{x}_1, \vec{x}_2) = (\vec{y}_1, \vec{y}_2)$ . We then show that for the remaining sequences  $\vec{K}$  not in  $\overline{KS}$  that the chances that  $O_{\vec{K}}(\vec{x}_1, \vec{x}_2) = (\vec{y}_1, \vec{y}_2)$  holds is small using the Unique Paths Lemma (Lemma 6).

We bound the size of  $\overline{KS}$  with high probability. In order to do so we will condition on event  $\overline{\mathbf{E}}$  from Claim 3. 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 key sequences  $\vec{\kappa} = (\kappa_1, \dots, \kappa_{m(n)})$  and  $\vec{\kappa}' = (\kappa'_1, \dots, \kappa'_{m(n)})$  where for some  $a$  we would have  $y_a = O_{\vec{\kappa}}(x_a) = O_{\vec{\kappa}'}(x_a)$ , and letting  $j$  be the largest index where  $\kappa_j \neq \kappa'_j$  this implies  $O_{\kappa_j}(O_{\kappa_1, \dots, \kappa_{j-1}}(x_a)) = O_{\kappa'_j}(O_{\kappa'_1, \dots, \kappa'_{j-1}}(x_a))$  contradicting the event  $\overline{\mathbf{E}}$ . We also condition on  $\overline{\mathbf{F}}$  from Claim 2 to ensure that responses from queries to  $O$  are independent of responses from queries to  $f$ .

Consider the probability that for any key sequence  $(k_1, \dots, k_{m(n)}) \in \overline{KS}$  that  $(\vec{y}_1, \vec{y}_2) = O_{k_1, \dots, k_{m(n)}}(\vec{x}_1, \vec{x}_2)$ . For each such sequence in  $\overline{KS}$  there exists an  $i$  s.t.  $(x_i, y_i) \notin Q_{k_1, \dots, k_{m(n)}}$  (otherwise  $|\mathcal{K}| \geq 1$  and  $\overline{G}$  would not output on line 8). Consider the smallest  $j$  such that there exists a  $b$  where  $(x_i, b) \in Q_{k_1} \circ \dots \circ Q_{k_j}$  but for all  $b'(x_i, b') \notin Q_{k_1} \circ \dots \circ Q_{k_j}$ . The probability that  $O_{k_j}(b) = O_{k_{j+1}, \dots, k_{m(n)}}^{-1}(y_i)$  is less than  $1/(2^n - |Q_{k_j}|) \leq 1/(2^n - i \cdot m(n))$ . Therefore, the probability there exists a key  $k \in \overline{KS}$  such that  $O_k(\vec{x}) = \vec{y}$  is less than  $\frac{\ell(n)+2}{(2^n - i \cdot m(n))}$ .

For the set of keys not in  $KS = \{0, 1\}^{m(n) \times n} \setminus \overline{KS}$  the Unique Paths Lemma shows that the probability that there exists a key  $\vec{K} \in KS$  such that  $O_{\vec{K}}(\vec{x}_1, \vec{x}_2) = (\vec{y}_1, \vec{y}_2)$  is no more than  $2^{n \cdot (m(n) - \ell(n) + 2)}$ .

Therefore, the probability of the claim is bounded by  $\frac{2(i \cdot m(n))^2}{2^n} + \frac{2i}{2^n} + \frac{(\ell(n)+2)}{(2^{n-i \cdot m(n)})} + 2^{n \cdot (m(n) - \ell(n) + 2)} < 2^{-n/2}$  (for sufficiently large  $n$ ), where the first two summands bound the probabilities of events **E** and **F** respectively.

...□

**Claim 6** *The probability that  $\overline{G}$  gives incorrect output on line 11 is less than  $2^{-n/2}$ .*

**Proof:** We observe that in order for  $|\mathcal{K}| > 1$  to occur, the event **E** of Claim 3 must occur at least  $\ell(n) + 2$  times: once for each  $y \in (\vec{y}_1, \vec{y}_2)$ . Therefore, we can use the bound on the probability of **E** to bound the probability of incorrect output by  $\frac{2(i \cdot m(n))^2}{2^n} < 2^{-n/2}$  (for sufficiently large  $n$ ).

...□

**Claim 7** *The probability that  $\overline{G}$  gives incorrect output on line 13 is less than  $\frac{1}{2^{n/2}}$ .*

**Proof:** We begin by conditioning on  $\overline{\mathbf{F}}$ , so that responses of queries to  $O_k$  ( $k \in \{0, 1\}^n$ ) are independent of the responses of queries to  $f$ . Let  $\vec{x}_1 = (x_1, \dots, x_{\ell(n)})$  and  $\vec{y}_1 = (y_1, \dots, y_{\ell(n)})$ . Let  $Q_f$  be the set of query response pairs made to  $f$ .

We consider two cases:  $\widehat{A}_{i+1}$ 's queries to  $f$  are not dependent on its query  $R_2(\vec{x}_1, \vec{y}_1)$  or  $\widehat{A}_{i+1}$  made at least one query to  $f$  dependent on its query  $R_2(\vec{x}_1, \vec{x}_2)$ .

In the first case we bound the probability that when  $R_2$  was queried that  $x_{\ell(n)+1}$  and  $x_{\ell(n)+2}$  had already been queried by any key (i.e.  $(x_{\ell(x)+2}, a) \in Q_i$  for some  $i$ ). We show the probability of this event is small. Assume the the query to  $f$  was the  $j$ th query, then the probability that there exists a  $\beta, k \in \{0, 1\}^n$  such that  $(x_{\ell(x)+1}, \beta) \in Q_k$  or  $(x_{\ell(x)+2}, \beta) \in Q_k$  is less than  $\frac{2 \cdot j \cdot m(n)}{2^n}$ . Next, we condition on that event not happening, and show that the probability that we will make any future queries  $O_k(a) = b$  where there exists a  $c$  such that  $(b, c) \in Q_f$  is small. This probability can easily be bounded by  $\sum_{s=j}^{i+1} \frac{|Q_f|}{2^{n-j \cdot m(n)}}$ . Therefore, the probability of the first case is less than  $\frac{2 \cdot j \cdot m(n)}{2^n} + \sum_{s=j}^{i+1} \frac{|Q_f|}{2^{n-s \cdot m(n)}} \leq \frac{2 \cdot i \cdot m(n)}{2^n} + \frac{(i+1) \cdot p(n)}{2^{n-(i+1) \cdot m(n)}} \leq 2^{-2n/3}$ .

In the second case when the adversary queries  $f$  it has already queried  $R_2$ , and therefore it is attempting to find a key  $\kappa = (\kappa_1, \dots, \kappa_{j-1}, f, \kappa_{j+1}, \dots, \kappa_{m(n)})$  such that  $(\vec{x}, \vec{y}) \in Q_\kappa$ . This can only happen if  $(a, y_t) \in Q_{f, \kappa_{j+1}, \dots, \kappa_{m(n)}}$  for some  $t$ . Fortunately, the probability of this occurring is small. It is unlikely that for any  $(a, b)$  in  $Q_f$  there will exist a  $(b, c) \in Q_k$  at the time the queries to  $f$  are made. Likewise it is unlikely that for queries  $a$  to  $f$  that  $f(a) \in \vec{y}$ , nor that any queries  $O_k(a) \in \vec{y}$  for any queries made to  $O$  after the queries to  $f$ . More formally, assume the query to  $f$  is the  $j$ th 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  $\frac{i}{2^{n-(j \cdot m(n))}}$ . The probability that any queries to  $O$  and  $R_3$  after the query to  $f$  will collide with  $\vec{y}$  is less than  $\frac{m(n) \cdot i}{2^{n-(i+1) \cdot m(n)}}$ . Therefore, the probability of the second case is bound by  $\frac{2i \cdot m(n)}{2^{n-(i+1) \cdot m(n)}}$ .

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

...□

**Claim 8** *The probability that  $\overline{G}$  gives incorrect output on line 16 is less than  $1/2^{n/2}$ .*

**Proof:** This is an application of Claim 3 and then an application of the Unique Paths Lemma. In particular, assuming  $\overline{\mathbf{E}}$  holds then our sets  $Q$  satisfy the requirements for the Unique Paths Lemma, where we're interested in all keys except the one in  $\mathcal{K}$  from  $\overline{G}$ . Therefore, by the Unique Paths Lemma we can bound the probability by  $2^{n \cdot (m(n) - \ell(n) + 2)}$ , 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  $2^{-n/2}$ .

...□

To finish proving Lemma 3 we simply take the union bound on the probability of errors in all of the claims, and this is less than  $5/2^{n/2}$  proving the lemma.

...□

### 4.3 Proof of Lemma 4

For the convenience of the reader we restate Lemma 4, but in this case we specify  $\delta(n) = \frac{5}{2^{n/2}}$ .

**Lemma 8** *For all sufficiently large  $n$ :*

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

**Proof:** We note that this proof is basically the same as the proof of Lemma 3 in the previous section. The only portion of the proof of Lemma 3 that relied on the fact that  $f \in O$  as opposed to  $f \in \Pi$  was Claim 2, which define 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 that in case  $\mathbf{F}$  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 4 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 3 which is  $5/2^{n/2}$ .

...□

## 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, 14], 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.

## References

- [1] W. Aiello, M. Bellare, G. Di Crescenzo, and R. Vekatesan. Security amplification by composition: The case of doubly-iterated, ideal ciphers. In H. Krawczyk, editor, *Advances in Cryptology - Crypto 98*, volume 1462 of *LNCS*, pages 390–407. Springer-Verlag, 1998.
- [2] B. Barak. How to go beyond the black-box simulation barrier. In IEEE, editor, *42nd IEEE Symposium on Foundations of Computer Science: proceedings: October 14–17, 2001, Las Vegas, Nevada, USA*, pages 106–115, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2001. IEEE Computer Society Press.
- [3] B. Barak, O. Goldreich, S. Goldwasser, and Y. Lindell. Resetably-sound zero-knowledge and its applications. In IEEE, editor, *42nd IEEE Symposium on Foundations of Computer Science: proceedings: October 14–17, 2001, Las Vegas, Nevada, USA*, pages 116–125, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2001. IEEE Computer Society Press.
- [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 (STOC)*, pages 103–112, Chicago, IL USA, May 1988. ACM Press.
- [5] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In IEEE, editor, *41st Annual Symposium on Foundations of Computer Science: proceedings: 12–14 November, 2000, Redondo Beach, California*, pages 305–313, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2000. IEEE Computer Society Press.
- [6] Rosario Gennaro, Yael Gertner, and Jonathan Katz. Bounds on the efficiency of encryption and digital signatures. Technical Report 2002-22, DIMACS, May 7 2002. Fri Jul 12 15:51:37 EDT 2002.
- [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: proceedings: 12–14 November, 2000, Redondo Beach, California*, pages 325–335, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2000. IEEE Computer Society Press.
- [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: proceedings: October 14–17, 2001, Las Vegas, Nevada, USA*, pages 126–135, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2001. IEEE Computer Society Press.
- [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. *JACM*, 38(3):690–728, July 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 (STOC)*, pages 291–304, Providence, RI USA, May 1985. ACM Press.
- [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, 1989.

- [13] Jeong Han Kim, D. R. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. In IEEE, editor, *40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 535–542, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1999. IEEE Computer Society Press.
- [14] Jeong Han Kim, D. R. Simon, and P. Tetali. Limits on the efficiency of one-way permutation-based hash functions. In IEEE, editor, *40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 535–542, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1999. IEEE Computer Society Press.
- [15] 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.
- [16] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17:373–386, 1988.
- [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, Milwaukee, Wisconsin, 23–25 October 1995. IEEE.
- [18] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In IEEE, editor, *40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 543–553, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1999. IEEE Computer Society Press.
- [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.