Lower Bounds on Black-Box Constructions of Pseudorandom Functions

Bar Alon*
alonbar08@gmail.com

Itai Dinur†
dinuri@bgu.ac.il

Muthuramakrishnan Venkitasubramaniam‡
textttmv783@georgetown.edu

June 7, 2026

Abstract

In their seminal work, Goldreich, Goldwasser, and Micali [CRYPTO 1984] constructed a pseudorandom function (PRF) using a black-box access to a pseudorandom generator (PRG). When combined with Levin's domain extension technique, the GGM construction invokes the PRG $\omega(\log n)$ times, where n denotes the input length to the PRG. To this day, no black-box construction achieving fewer calls is known.

Recently, Beimel, Malkin, and Mazor [CRYPTO 2024] showed that for a certain family of constructions, which they termed *tree constructions*, the GGM construction is optimal. However, the basic challenge of whether a PRF can be built with just *one invocation* of the PRG still remains open.

In this work, we consider fully black-box constructions of PRFs from PRGs, where both the construction and the reduction are required to be black-box, and the number of interactions the reduction makes with the adversary is independent of the number of oracle calls the adversary makes to its underlying function within each interaction. Our main result shows that no such construction can have $o(n/\log n)$ and $o(\mathsf{in}/\log \mathsf{in})$ *non-adaptive* calls to the PRG, where in is the input length of the PRF. This impossibility holds even for weak PRFs with one-bit output, where the adversary is restricted to making i.i.d. uniformly random queries. In addition, we prove a lower bound for weak PRFs with sufficiently long outputs that holds even when the construction is allowed to make adaptive queries to the PRG.

*School of Electrical Engineering, Tel-Aviv University. Work was conducted while at Georgetown University.

†Department of Computer Science, Ben-Gurion University and Department of Computer Science, Georgetown University

‡Department of Computer Science, Georgetown University

Contents

1 Introduction **1**
1.1 Our Results . . . 1
1.2 Related Works . . . 3
1.3 Organization . . . 3

2 Our Techniques **3**
2.1 Lower Bounds for Fully Black-Box Constructions . . . 3
2.2 Lower Bounds for PRFs With Long Outputs . . . 8

3 Preliminaries **8**
3.1 Notations . . . 8
3.2 Pseudorandom Generators . . . 9
3.3 Black-Box Constructions of Pseudorandom Functions . . . 10

4 Lower Bounds for Fully Black-Box Constructions of PRFs From PRGs **11**

5 Lower Bounds for Black-Box Constructions of a PRF With a Long Output **24**

Bibliography **26**

1 Introduction

A pseudorandom function (PRF) family is a collection of functions that appear random to any efficient algorithm. PRFs are fundamental to cryptography, both in theory and in practice, and have found many applications in areas outside of cryptography, such as in complexity and learning theory.

The seminal work of Goldreich, Goldwasser, and Micali [9] (henceforth the GGM construction) shows that a PRF can be constructed from *any* pseudorandom generator – an efficient algorithm that on a random input, outputs a longer string that appears random to any efficient algorithm. Specifically, they showed that given black-box access to a pseudorandom generator (PRG) $G : \{0,1\}^n \to \{0,1\}^{2n}$, one can construct a PRF $F = \left\{ f_k : \{0,1\}^{\mathsf{in}(n)} \to \{0,1\}^n \right\}_{k \in \{0,1\}^n}$ for any polynomial $\mathsf{in}(n)$. The number of calls to G in the GGM construction is $|x|$ for any input x. A variant of the GGM construction that relies on Levin's domain extension technique [12] improves the number of calls to $\omega(\log n)$.[1] However, a fundamental question has remained open since it was first discovered—is the construction optimal in the number of PRG invocations? More precisely, the following remains open:

What is the minimal number of invocations of a (length-doubling) PRG required to build a PRF?

Recently, Beimel et al. [2] made progress in answering this question. For a limited class of constructions that they refer to as *tree constructions* and generalizes the GGM construction, they prove that the GGM construction is optimal. In fact, their result holds even when considering PRFs with one-bit output. However, the state of affairs for more general (i.e. not necessarily tree) constructions, still remains open. In fact, previous results do not rule out the possibility of building a PRF with a *single call* to the PRG.

1.1 Our Results

We take a step towards answering the above question. We show two results. Our first and main result considers the already challenging setting of designing PRFs via non-adaptive calls to the underlying PRG (i.e., the construction generates all of its queries to the PRG before sending them). We demonstrate a lower bound on the number of calls to the underlying PRG oracle required for a fully black-box construction of a PRF. The impossibility is for a class of black-box reductions, which we call *query-bounded* reductions, which we define below. For our second result, we provide a lower bound for PRFs with a long output, which additionally includes adaptive constructions.

Lower bounds for black-box constructions with non-adaptive calls. Our main result demonstrates the impossibility of building a PRF with input length in from a PRG with an arbitrary polynomial stretch, using c *non-adaptive* calls, where $c = o(n/\log n)$ and $c = o(\mathsf{in}/\log \mathsf{in})$ via a fully black-box construction. We demonstrate this impossibility for a class of black-box reductions, which we call *query-bounded* reductions.

Roughly, a reduction is query-bounded if the number of interactions it makes with the adversary is upper-bounded by a polynomial that is independent of the adversary's query complexity (for all

[1] The final construction adds to the key a hash $h : \{0,1\}^{\mathsf{in}(n)} \to \{0,1\}^{\omega(\log n)}$ from pair-wise independent hash function family, and applies the GGM construction to $h(x)$.

sufficiently successful adversaries). Recall that in the PRF setting, the adversary has access to a function oracle that it may call a bounded number of times. We focus on reductions whose number of interactions with the adversary may depend on the adversary's success probability, but is independent of the number of oracle calls the adversary makes to the underlying function within each interaction. We believe this is a natural class of reductions and captures all reductions in the literature (see Section 2.1.3). In particular, it includes reductions whose number of interactions with the adversary decreases as the adversary's success probability increases. It is an intriguing question whether one could extend the lower bound to all black-box reductions.

Theorem 1.1 (Informal, impossibility for fully black-box construction)**.** *There is no fully black-box construction of a PRF with input length* in *from a PRG using query-bounded reductions and* c *non-adaptive calls, where* $c = o(n/\log n)$ *and* $c = o(\mathsf{in}/\log \mathsf{in})$*.*

The formal statement and proof of Theorem 1.1 appear in Section 4. A few notes are in place. First, our proof works even for *weak* PRFs, where the adversary can only make i.i.d. uniform queries to the oracle. Second, unlike our second result (see Theorem 1.2 below), Theorem 1.1 holds for PRFs with one-bit output. Finally, we argue that as a corollary of Theorem 1.1, there is no fully black-box construction of a PRF from a PRG using a query-bounded reduction and a *constant number* of non-adaptive calls, regardless of the input length of the PRF. Indeed, for $\mathsf{in} = O(\log n)$ PRFs can be constructed unconditionally by simply letting the key be the truth table of the function (in particular, there is no reduction). On the other hand, for $\mathsf{in} = \omega(\log n)$, Theorem 1.1 states that there is no fully black-box construction with a constant number of calls. Note that this resolves the question of the single call case for query-bounded reductions.

A lower-bound for black-box constructions with long outputs. We show an additional lower bound for PRFs with long outputs. Unlike in Theorem 1.1, here the result holds for adaptive constructions and does not require assuming the reduction to be query-bounded.

Theorem 1.2 (Informal, lower bounds for long output constructions)**.** *There is no fully black-box construction of a PRF with output length* out *from a PRG with stretch* r *using at most* $\frac{\mathsf{out}}{\max(r,\omega(\log n))}$ *calls.*

Stated differently, if $r = O(\log n)$, then Theorem 1.2 states that there is no black-box construction using $o(\mathsf{out}/\log n)$ calls to the PRG. On the other hand, if $r = \omega(\log n)$, then there is no black-box construction using $o(\mathsf{out}/r)$ calls. Additionally, similarly to Theorem 1.1, the proof of Theorem 1.2 also applies to weak PRFs.

Let us compare Theorem 1.2 to the GGM construction (note that in the GGM construction, the output of the PRF is of length n, making it natural to compare to). The GGM construction (combined with Levin's trick) is an n-bit output PRF from an n-bit stretch PRG using $\omega(\log n)$ calls to the PRG. The PRG can be constructed from a 1-bit stretch PRG using $O(n/\log n)$ calls using the Goldreich-Levin hardcore function theorem [8]. Thus, the overall number of calls the PRF construction makes to a 1-bit stretch PRG is $\omega(n)$. By Theorem 1.2, the number of calls must be at least $n/\omega(\log n)$. This means that the GGM construction is optimal up to a factor of $\omega(\log n)$. The formal statement of Theorem 1.2 appears in Section 5.

1.2 Related Works

Separation of cryptographic primitives is a fundamental question in cryptography that qualitatively measures the relative hardness of different cryptographic primitives. The seminal work of Impagliazzo and Rudich [11] provided the first separations ruling out constructions of key exchange from one-way functions. The class of constructions ruled out in this work is referred to as a fully black-box construction (see [17] for a taxonomy on reductions) where the construction uses the underlying primitive as a black-box and the reduction is black-box.

Within the framework of black-box separations, the *meta-reduction* paradigm introduced in the work of Boneh and Venkatesan [4], considers designing a (possibly inefficient) attacker $\mathcal{A}$ that when invoked with the reduction breaks the challenge in the underlying security reduction, then emulating $\mathcal{A}$ in a manner that is efficient and does not affect the interaction of the reduction and the challenger (see also [5, 10, 6, 15, 16]). We follow the meta-reduction paradigm in this work (see [1] and references therein).

Miles and Viola [13, 14] were the first to consider lower bounds on black-box constructions of PRFs from PRGs.[2] They ruled out single-call constructions where the output is one of the PRG's output bits. More recently, Beimel et al. [2] both generalized the result of [13, 14], and proved a lower bound on the number of calls in tree constructions, which naturally generalize the GGM construction. Additionally, they showed a lower bound on the key length required for black-box constructions.

1.3 Organization

We give an overview of our techniques in Section 2. The preliminaries are given in Section 3. Then, in Section 4 we prove our main result, showing a lower bound for constructions with non-adaptive calls. Finally, in Section 5, we prove our lower bound for PRFs with long outputs.

2 Our Techniques

We now present our techniques. We start with showing the impossibility for non-adaptive black-box constructions.

2.1 Lower Bounds for Fully Black-Box Constructions

As a warm-up, we will rule out PRF constructions that make a single call to a PRG and then generalize this argument for multiple non-adaptive calls.

2.1.1 Warm-Up: The Single Call Case

Since we assume the construction makes one call to the PRG, we may write it as

$$f_k^G(x) = P\left(x, k, G(S(x, k))\right),$$

for efficiently computable functions S and P. In the following, let λ denote the length of the keys, and let r denote the stretch of G.

[2] The paper considers the question of extending the stretch of a PRG; however, they noted in [14] that their result applies to constructing PRFs from PRGs as well.

Standard black-box separation like the approaches in [13, 2] construct an oracle $\mathcal{O}$ relative to which a PRG G exists, and construct an oracle-aided algorithm $\mathcal{A}^{\mathcal{O}}$ that breaks the security of F using a polynomial number of calls to $\mathcal{O}$. We argue that it may be impossible to use such an approach to rule out black-box constructions of PRF from a PRG. Consider the following PRF construction from an r-bit stretch PRG G:

$$f^G_{h,k}(x) = \langle G(h(x)), k\rangle$$

where the key is a pair (h, k), where h is sampled from a pairwise independent hash function family, $k \leftarrow \{0,1\}^{n+r}$, and $\langle\cdot,\cdot\rangle$ is the inner product computed modulo 2. If G is instantiated via a random oracle, then, for any polynomial set of inputs $x_1, \ldots, x_m$, the set $\{h(x_1), \ldots, h(x_m)\}$ will all be distinct (except with negligible probability). This would imply that the entropy of each $G(x_i)$ is large enough (given the other outputs of f) and the inner product can extract a random bit, such that the m bits are statistically close to a uniform sample from $\{0,1\}^m$. In fact, this argument works even if G is only "mildly" random, e.g., G applies a random function only to the first $\omega(\log n)$ bits. Since the proof of security of the PRF is information-theoretic and does not depend on the reduction, it may be impossible to rule out such constructions using random oracles and an adversary $\mathcal{A}^{\mathcal{O}}$ that calls $\mathcal{O}$ a polynomial number of times. Since the work of Gennaro and Trevisan [7], most black-box separations from a random oracle have pursued this approach.

We circumvent this challenge by relying on the meta-reduction paradigm. We first define an "ideal" adversary that breaks the PRF using an *exponential number* of queries to the PRG. We then construct an *inefficient* "real" adversary that can simulate the ideal adversary when given the queries made by the reduction to the PRG. The idea here is that such an adversary can be simulated together with the reduction. In other words, we construct an algorithm that breaks the security of the PRG by emulating the reduction with real adversary that together make only polynomial queries to the underlying PRG.

A naive attempt. Let us start with a naive construction for the real and ideal adversaries that *does not* work. Define the ideal adversary $\mathcal{A}^{G,f}_{\mathsf{ideal}}$ to query G on all inputs, and query the PRF/random function f on m distinct inputs $x_1, \ldots, x_m$. The adversary outputs 1, indicating PRF, if and only if it can find a key that is consistent with all the responses. Clearly, if f is sampled from the PRF family, $\mathcal{A}_{\mathsf{ideal}}$ will always output 1. When f is sampled from the random function family, we can show that it will output 1 with negligible probability, assuming m is sufficiently large via an entropy argument. Since the key length is λ, if $m = \lambda + n$, then except with probability 2^{-n}, there will not be a consistent key.

Now, consider the following "real adversary" that aims to emulate the ideal adversary. This adversary queries its oracle (whose answers are given by reduction) on m inputs just like the ideal adversary, but along with the responses from the reduction, it also receives all queries made by the reduction to the PRG. The idea is that in order for the responses to be consistent with the PRF family, the reduction must have queried the PRG on all the points needed to emulate the PRF for some key k. Thus, using the query-answers provided to the real adversary, we let it (in exponential time) find the key k without making any additional queries to the PRG itself. If it finds such a key, the adversary outputs 1 indicating PRF and outputs 0 otherwise.

This adversary seemingly emulates the ideal adversary. However, this is not the case. The reduction does not have to make all the queries to the PRG consistent with a particular key to compute the answers. In fact, in the GGM reduction, the adversary implants the challenge string

as the output of a PRG call. This means that the above strategy for the real adversary fails, because the reduction may have computed some of the outputs using some key and querying G, but guessed the other outputs (without querying G). Therefore, the real adversary cannot find a consistent key, while the ideal adversary may find one.

Our first observation is that the reduction cannot guess too many of the outputs, while still being consistent with some key. Indeed, otherwise, the probability that it is consistent with some key becomes negligibly small. Thus, we can revise the strategy of the two adversaries to only look for a key that is consistent with, say, 0.8 fraction[3] of the inputs. Indeed, when using 0.8 as the threshold, if the ideal adversary interacts with a random function f, then for every key, on expectation, only 1/2 of the answers are consistent with the inputs. By a simple concentration bound, it will not find a key except with negligible probability.

Unfortunately, there is the issue of the reduction guessing too *few* of the outputs. Specifically, in the case where the reduction computed correctly only 0.7 fraction of the inputs, we do not know how to show that the ideal adversary will not find a *different* key that is consistent with 0.8 fraction of the inputs.

We introduce two ideas to solve this. Towards describing our ideas, we first introduce two *score functions* (one for each adversary). For the ideal adversary, the score function with respect to some key is simply the number of consistent input-output pairs. The real adversary estimates the score function of the ideal adversary by considering the probability that it will be consistent with the answers (instead of a binary value). Specifically, for every key k the real score is computed as follows: for every x that is one of the m inputs (and $S(x,k)$ has not been queried by the reduction), compute the probability over a random G' that is consistent with the reduction queries, that $f_k^{G'}(x)$ equals the value given to it by the reduction. This probability can be computed exactly by the real adversary. The final score of the real adversary for k is the sum of these probabilities over all inputs. By linearity of expectation, it is equal to the expected score of the ideal adversary (for the same key). The final score of each of the real and ideal adversaries is its maximal score over all keys, and the answer (PRF/random function) is defined by comparing this score to some threshold.

However, choosing a fixed threshold could fail in case it falls between the real and ideal scores (which could happen even if the scores are very close). Therefore, the second idea is to choose a *random threshold* between 0.8 and 1. This helps to smoothen the difference and as long as the real and ideal scores are close enough.

However, there are still two issues with the above argument.

1. Although for every k, the score of the real adversary equals the expected score of the ideal adversary, the final outputs of the PRF on the m inputs might be dependent. Hence, we cannot claim that the scores are close with high probability. For a key k, such dependencies occur if for two different inputs x_i and x_j, $S(x_i,k) = S(x_j,k)$.

2. The second issue relates to the challenge y^* held by the reduction. Recall that y^* is either uniform over $\{0,1\}^{n+r}$, or equals to $G(s)$, where $s \leftarrow \{0,1\}^n$. In the latter case, if the reduction implant y^* instead of $G(S(x_i,k))$ for some i, then this increases the ideal score by 1, but increases the real score only by a negligible amount. Otherwise, the challenge affects the scores the same, which may provide the reduction a way to distinguish between the two adversaries.

[3]The choice of 0.8 is arbitrary; any constant larger than 1/2 and smaller than 1 works.

Fixing the issues. To solve the problems, we aim to break the dependencies between the outputs of the PRF. This solves the first issue as it allows us to use Chernoff's inequality to argue that the two scores are close with high probability. It also solves the second one since if the outputs are independent, the challenge can only affect at most one output (keeping the scores close). Towards breaking the dependencies, we observe that for every key k, either there is a PRG query that appears for many inputs, or we can find a large set of inputs with distinct PRG seeds.

Claim 2.1 (Informal)**.** *For a positive integer parameter p (to be chosen), one of the following must hold for every key k.*

1. *There exists a seed that appears at least p times in $\{S(x_i, k) : i \in [m]\}$, or*
2. *There is a set $\mathcal{I} \subseteq [m]$ of at least m/p inputs with distinct PRG seeds.*

Proof. Consider the graph $H = ([m], E)$, where i, j is an edge if and only if $S(x_i, k) = S(x_j, k)$. If Item 1 does not hold, then the maximum degree in H is at most $p - 1$. By greedily selecting the vertices, we conclude there is an independent set of size at least m/p. By definition, the PRG seeds associated with an independent set are distinct. □

Now, if for a key k the second item holds, then our previous argument works when restricting to $\mathcal{I}$: Since we instantiate the PRG with a random oracle, this implies that the corresponding outputs are independent random variables (over the PRG random oracle). This independence allows us to use a concentration bound to argue that if m/p is sufficiently large, then the real adversary can predict the output of the ideal adversary with good accuracy. As for the other keys, we argue that there is only a negligible fraction of them. Indeed, if for too many keys there is a seed s that appears at least p times then, intuitively, for a random key the entropy in the output for those p inputs is bounded by $\lambda + n + r$. Thus, for a sufficiently large p, the PRF can be distinguished from random. As a result, we may allow both the real and ideal adversaries to *ignore* such keys.

With this, for any polynomial u, we can choose m so that for every call made by the reduction to the adversary, the statistical gap between the real and ideal adversary's outputs is $1/u$-close. However, recall that the reduction can call the adversary polynomially many times, say v. If we want the argue that the statistical gap for all the combined outputs are close we need v/u to be small. In other words u should be chosen after v. Since m depends on u, for our impossibility to hold, we need to assume that the number of calls made by the reduction v is independent of m. We believe this is a reasonable restriction and discuss this further in Section 2.1.3.

2.1.2 The General Case

We now describe our ideas to generalize the single call case to c non-adaptive calls, where $c = o(n/\log n)$ and $c = o(\mathsf{in}/\log \mathsf{in})$. Recall that to define the two adversaries, we define two score functions (one for each adversary) that aim to compute for how many inputs the reduction returned the correct output (with respect to some key). The two score functions are defined with respect to each key, via a procedure that depends only on the construction of the PRF (in particular, it is independent of the output of the PRG). This is important, since the score function does not require the real adversary to make any PRG queries.

As in the single call case, the goal of the functions is to find a large set of inputs that make disjoint PRG queries. At a high level, this is done by letting the score functions try to break the

dependence between some PRF outputs by fixing PRG outputs on queries that are made by many PRF inputs. Specifically, the procedure works in iterations as we next describe.

In each iteration, the adversaries looks for a seed s that appears for sufficiently many inputs (i.e., for many of the x_i's, at least one of the seeds needed to compute $f_k^G(x_i)$ is s). For such seeds, the PRG will not contribute much entropy to the output of the PRF (taken over all inputs that query s). When we define the scores, we replace $G(s)$ with a fixed value that maximizes the score and is *independent* of G. In such a case, the construction is left with one less call to the PRG. This process continues until either no s appear many times, or until the process finds c such seeds. In the former case, we can apply (a generalization of) Claim 2.1 to argue that there is a large set of inputs with a mutually disjoint set of seeds, while in the latter case, we will later replace all calls to the PRG with fixed values.

In more detail, for every key k, the final output of the procedure is a list of $\ell(k) \leq c$ seeds $s_1, \ldots, s_{\ell(k)}$ and a list of nested sets of inputs $\mathcal{I}_1 \supseteq \ldots \supseteq \mathcal{I}_{\ell(k)}$. We require that each s_t appears for sufficiently many inputs in $\mathcal{I}_t$. The score calculation is then done by replacing each $G(s_t)$ with a fixed value. If $\ell(k) = c$, then the scores are fixed and independent of G. Otherwise, there is a set $\mathcal{I} \subseteq \mathcal{I}_{\ell(k)}$ of inputs with mutually disjoint seeds. The score calculation is then done the same as in the single call case.

For the argument to work, we need to ensure that at every step, the set of inputs $\mathcal{I}_t$ that are being considered is sufficiently large. Specifically, if the procedure stops early, then Claim 2.1 implies the existence of a set of $\frac{|\mathcal{I}_t|}{(c-t)|\mathcal{I}_{t+1}|}$ inputs with mutually disjoint queries. For the concentration bound to work, this ratio needs to be sufficiently large (Specifically, it needs to be larger than some fixed polynomial that depends on the key length, the PRG stretch, and the reduction). If the procedure replaces every $G(s_t)$, we need $|\mathcal{I}_{\ell(k)}|$ to be sufficiently large as well to argue that the PRF can be distinguished from a random function. This implies that the number of inputs required is $m = (\text{poly}(n))^c \cdot c!$. For the adversaries to be well-defined, this value must be upper bounded by 2^{in}, which holds for all sufficiently large n's by our assumption that $c = o(\mathsf{in}/\log \mathsf{in})$. Additionally, the total number of queries that R makes to the PRG is $d \cdot m^\beta$, for some constant $\beta \in \mathbb{N}$ and where d is the number of calls R makes to the distinguishing adversary. Recall that $c = o(n/\log n)$, hence $d \cdot m^\beta \leq 2^{n/4}$ for all sufficiently large n's. Since we instantiate the PRG with a random function, $2^{n/4}$ queries are insufficient to break its security (see Lemma 3.3 for the formal statement), resulting in a contradiction.

2.1.3 Subtleties With Oracle Access to PRF Distinguishers

The standard definition of a fully black-box reduction from PRF to PRG is as follows. Given access to an oracle-adversary $\mathcal{A}^{(\cdot)}$ that is able to distinguish the PRF from a random function, the reduction, a PPT oracle-aided algorithm R is able to distinguish the output of $G(s)$, where $s \leftarrow \{0,1\}^n$, from a string uniformly sampled from $\{0,1\}^{|G(s)|}$.

Since we fixed R, its run-time is some fixed polynomial, say $T(n)$. The adversary $\mathcal{A}^f$ is allowed to query f a polynomial m number of inputs. When activated by the reduction, $\mathcal{A}$ expects $m(n)$ answers. These should be provided by R; however, m is chosen by $\mathcal{A}$, which is fixed after R, hence m cannot depend on T. In particular, if $m(n) > T(n)$, then R has no time to answer $\mathcal{A}$'s queries.

For the model to make sense, we have to treat the m inputs chosen by $\mathcal{A}$ as part of the reduction's input. However, now the reduction depends on $\mathcal{A}$. For example, after receiving the m inputs from $\mathcal{A}$, the reduction can decide to call it m more times. If we now replace $\mathcal{A}$ with another adversary that sends $m+1$ inputs, and apply $\mathcal{A}$ on the first m answers (i.e., it ignores the last input and

output), then R calls it more times than with $\mathcal{A}$, thus the entire procedure changed, even though the adversary does the same. Such reductions are arguably not "black-box".

Recall that in our proof, we had to assume that the reduction cannot depend on the number of inputs given by the distinguishing oracle. This means that our proof does not hold for the standard notion of fully black-box constructions. Instead, we consider *query-bounded* reductions. These are reductions whose number of calls to the adversary is bounded by some polynomial $d(n)$ for all adversaries whose distinguishing advantage is sufficiently large. Specifically, for our proof, we assume that there exists a polynomial $d(n)$ such that for any adversary $\mathcal{A}$ whose distinguishing advantage is at least $\alpha(n) = 1 - e^{-n}$, the reduction calls $\mathcal{A}$ at most $d(n)$ times. Note that any reduction whose number of calls to $\mathcal{A}$ *decreases* as a function of the distinguishing advantage of $\mathcal{A}$ is also query-bounded (perhaps with a different $\alpha(n)$). We argue that the class of query-bounded reductions is both natural and well-motivated. Indeed, all reductions previously considered in the literature, to the best of our knowledge, satisfy this property and thus fall within the class of query-bounded reductions. This includes, for example, the GGM construction [9], Goldreich and Levin [8] hardcore predicate construction, and Waters [18] construction of signature schemes. We refer the reader to Definition 3.6 for a formal definition of query-bounded reductions.

2.2 Lower Bounds for PRFs With Long Outputs

We now prove Theorem 1.2. To simplify this introduction, we will prove a simpler result that considers constructions with a single call to the PRG (see Section 5 for the proof of the general result). Although it is not as general, the proof highlights the key ideas of our techniques. Specifically, we show that if $\mathsf{out} - r = \omega(\log n)$, then there is no black-box construction of a PRF with output length out from a PRG with stretch r using a single call to the PRG.

Let $\mathsf{in} = \mathsf{in}(n)$ and assume that $F = \{f_k^G : \{0,1\}^{\mathsf{in}} \to \{0,1\}^{\mathsf{out}}\}_{n\in\mathbb{N},k\in\{0,1\}^\lambda}$ is a PRF construction that calls $G : \{0,1\}^n \to \{0,1\}^{n+r}$ once. The main idea of the proof is to consider the entropy of sufficiently many applications of the function. That is, given access to a function f (that is either truly random or sampled from F), we consider the total entropy of m applications of f on different inputs, where $m = m(n)$ is a sufficiently large polynomial. Observe that for a random function the entropy is maximized, i.e., it is $\mathsf{out} \cdot m$. However, we show how to construct a PRG G such that the entropy of m applications of f_k^G is noticeably far from $\mathsf{out} \cdot m$. Intuitively, this is achieved by considering a PRG that is only applied to the first $\mathsf{out} - r - 1 = \omega(\log n)$ bits of the input. This ensures that the entropy in the output comes from the key k and the $\mathsf{out} - 1$ (pseudo)random bits obtained from the stretch of G. Thus, the entropy will be at most $\lambda + m \cdot (\mathsf{out} - 1)$. Taking $m = \lambda + n$ results in a gap of n from $\mathsf{out} \cdot m$, which we show is sufficient.

We note that the above argument works for all fully black-box reductions, and not just query-bounded ones, as was needed in Theorem 1.1.

3 Preliminaries

3.1 Notations

For $n, m \in \mathbb{N}$ we let $[n] = \{1, 2 \dots n\}$ and let $\binom{[n]}{m} = \{\mathcal{S} \subseteq [n] : |\mathcal{S}| = m\}$. All logarithms are taken in base 2. For a set $\mathcal{S}$, we write $s \leftarrow \mathcal{S}$ to indicate that s is selected uniformly at random from $\mathcal{S}$. Given a random variable (or a distribution) X, we write $x \leftarrow X$ to indicate that x is selected according to X. PPT stands for probabilistic polynomial time. A function $\mu : \mathbb{N} \to [0,1]$ is called

negligible, if for every positive polynomial $p(\cdot)$ and all sufficiently large n, $\mu(n) < 1/p(n)$. For $n, m \in \mathbb{N}$, we let $\mathcal{F}_{n,m} = \{f : \{0,1\}^n \to \{0,1\}^m\}$ denote the set of all functions from $\{0,1\}^n$ to $\{0,1\}^m$, and let $\mathcal{F}_n = \mathcal{F}_{n,1}$.

Theorem 3.1 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_n \in \{0,1\}$ denote independent random variables such that $\Pr[X_i = 1] = p_i$ for all $i \in [n]$ for some $p_i \in (0,1)$. Let $p = \sum_{i=1}^n p_i$. Then for every $t > 0$,*

$$\Pr\left[\sum_{i=1}^n X_i \geq pn + t\right] \leq e^{-2t^2/n}$$

and

$$\Pr\left[\sum_{i=1}^n X_i \leq pn - t\right] \leq e^{-2t^2/n}.$$

3.2 Pseudorandom Generators

Definition 3.2 (Pseudorandom generator (PRG))**.** *An efficiently computable function G is an $r(n)$-bit stretch PRG if for every $n \in \mathbb{N}$ and $s \in \{0,1\}^*$, $|G(s)| = |s| + r(|s|)$ and for every* PPT *algorithm $\mathcal{A}$, there exists a negligible function μ such that*

$$\left|\Pr_{s \leftarrow \{0,1\}^n}[\mathcal{A}(1^n, G(s)) = 1] - \Pr_{y \leftarrow \{0,1\}^{n+r(n)}}[\mathcal{A}(1^n, y) = 1]\right| \leq \mu(n).$$

The following lemma states that a random function is a PRG with overwhelming probability.

Lemma 3.3. *The following holds for every $r : \mathbb{N} \to \mathbb{N}$, and all sufficiently large $n \in \mathbb{N}$. Let $G : \{0,1\}^n \to \{0,1\}^{n+r(n)}$ be a function chosen uniformly at random and C be a $2^{n/4}$-call oracle aided (possibly inefficient) algorithm. Then with probability at least $1 - 2^{-n/2}$,*

$$\left|\Pr_{s \leftarrow \{0,1\}^n}\left[C^G(G(s)) = 1\right] - \Pr_{y \leftarrow \{0,1\}^{n+r(n)}}\left[C^G(y) = 1\right]\right| \leq 2^{-n/4}.$$

Proof. By Markov's inequality, it suffices to show that

$$\mathrm{E}_{G \leftarrow \mathcal{F}_{n,n+r(n)}}\left[\left|\Pr_{s \leftarrow \{0,1\}^n}\left[C^G(G(s)) = 1\right] - \Pr_{y \leftarrow \{0,1\}^{n+r(n)}}\left[C^G(y) = 1\right]\right|\right] \leq 2^{-3n/4}.$$

This is done by lazy sampling [3]. Assume without loss of generality that C makes only distinct queries to the oracle. Let $P_0 : \{0,1\}^n \to \{0,1\}^{n+r(n)}$ be the procedure defined as follows:

1. Sample $s \leftarrow \{0,1\}^n$ and $y \leftarrow \{0,1\}^{n+r(n)}$, independently.
2. On query $x \in \{0,1\}^n$, sample $y^* \leftarrow \{0,1\}^{n+r(n)}$.
3. If $s = x$, then set $bad = \mathsf{true}$. Set $y = y^*$.
4. Return y.

Additionally, let $P_1 : \{0,1\}^n \to \{0,1\}^{n+r(n)}$ be the procedure defined as follows:

1. Sample $s \leftarrow \{0,1\}^n$ and $y \leftarrow \{0,1\}^{n+r(n)}$, independently.
2. On query $x \in \{0,1\}^n$, sample $y^* \leftarrow \{0,1\}^{n+r(n)}$.

3. If $s = x$, then set $bad = \mathsf{true}$.

4. Return y.

First, note that

$$\mathrm{E}_{G \leftarrow \mathcal{F}_{n,n+r(n)}} \left[\Pr_{s \leftarrow \{0,1\}^n} \left[C^G(G(s)) = 1 \right] \right] = \Pr_{y^* \leftarrow \{0,1\}^{n+r(n)}} \left[C^{P_0}(y^*) = 1 \right]$$

and that

$$\mathrm{E}_{G \leftarrow \mathcal{F}_{n,n+r(n)}} \left[\Pr_{y \leftarrow \{0,1\}^{n+r(n)}} \left[C^G(y) = 1 \right] \right] = \Pr_{y \leftarrow \{0,1\}^{n+r(n)}} \left[C^{P_1}(y) = 1 \right].$$

Therefore,

$$\begin{aligned} &\mathrm{E}_{G \leftarrow \mathcal{F}_{n,n+r(n)}} \left[\left| \Pr_{s \leftarrow \{0,1\}^n} \left[C^G(G(s)) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^{n+r(n)}} \left[C^G(y) = 1 \right] \right| \right] \\ &\leq \left| \Pr_{y^* \leftarrow \{0,1\}^{n+r(n)}} \left[C^{P_0}(y^*) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^{n+r(n)}} \left[C^{P_1}(y) = 1 \right] \right| \end{aligned}$$

Now, observe that the procedures P_0 and P_1 are identical-until-*bad*, namely, they return identically distributed answers until *bad* occurs. Since C can make $2^{n/4}$ queries, *bad* occurs with probability at most $2^{n/4-n} = 2^{-3n/4}$. Therefore,

$$\left| \Pr_{y^* \leftarrow \{0,1\}^{n+r(n)}} \left[C^{P_0}(y^*) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^{n+r(n)}} \left[C^{P_1}(y) = 1 \right] \right| \leq 2^{-3n/4}.$$

□

3.3 Black-Box Constructions of Pseudorandom Functions

We next define black-box constructions of a pseudorandom function (PRF) from PRGs.

Definition 3.4 (Fully black-box PRF constructions)**.** *A* fully black-box construction *of a PRF from an $r(n)$-bit stretch PRG is a pair (F, R) where*

$$F = \left\{ f_k^{(\cdot)} : \{0,1\}^{\mathsf{in}(n)} \to \{0,1\}^{\mathsf{out}(n)} \right\}_{n \in \mathbb{N}, k \in \{0,1\}^{\lambda(n)}}$$

is an efficiently computable oracle-aided function family, and $R^{(\cdot),(\cdot)}$ is a PPT *oracle-aided algorithm such that the following holds. There exists a polynomial $q : \mathbb{N} \to \mathbb{N}$ such that for every $G = \{G_n : \{0,1\}^n \to \{0,1\}^{n+r(n)}\}_{n \in \mathbb{N}}$ and every oracle-aided algorithm $\mathcal{A}^{(\cdot)}$ for which there exists a constant $c \in \mathbb{N}$ such that*

$$\left| \Pr_{k \leftarrow \{0,1\}^{\lambda(n)}} \left[\mathcal{A}^{f_k^G}(1^n) = 1 \right] - \Pr_{f \leftarrow \mathcal{F}_{\mathsf{in}(n),\mathsf{out}(n)}} \left[\mathcal{A}^f(1^n) = 1 \right] \right| \geq \frac{1}{n^c}$$

for infinitely many n's,

$$\left| \Pr_{s \leftarrow \{0,1\}^n} \left[R^{\mathcal{A},G}(1^n, G(s)) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^{n+r(n)}} \left[R^{\mathcal{A},G}(1^n, y) = 1 \right] \right| \geq \frac{1}{q(n^c)}$$

for infinitely many n's.

(F, R) is fully black-box construction of a weak PRF*, if the above holds for all algorithms $\mathcal{A}$ that query i.i.d. uniform queries to f.*

We next define the query complexity of a black-box construction.

Definition 3.5 (c-call adaptive and non-adaptive constructions)**.** *Let* $\mathsf{in}(n)$, $\mathsf{out}(n)$, $r(n)$, *and* $\lambda(n)$ *be polynomials, and let*

$$F = \left\{ f_k^{(\cdot)} : \{0,1\}^{\mathsf{in}(n)} \to \{0,1\}^{\mathsf{out}(n)} \right\}_{n \in \mathbb{N}, k \in \{0,1\}^{\lambda(n)}}$$

*be a black-box PRF construction from an $r(n)$-bit stretch PRG. For a function $c : \mathbb{N} \to \mathbb{N}$, we say that F is a $c(\cdot)$-*call construction *if for every $i \in [c(n)]$ there exist $S_i : \{0,1\}^{\mathsf{in}(n)} \times \{0,1\}^{\lambda(n)} \times (\{0,1\}^{n+r(n)})^{i-1} \to \{0,1\}^n$ and $P : \{0,1\}^{\mathsf{in}(n)} \times \{0,1\}^{\lambda(n)} \times (\{0,1\}^{n+r(n)})^{c(n)} \to \{0,1\}^{\mathsf{out}(n)}$ such that*

$$f_k^G(x) = P\left(x, k, G\left(s_1\right), \ldots, G(s_c)\right)$$

for all $G : \{0,1\}^n \to \{0,1\}^{n+r(n)}$, where $s_i = S_i(x, k, G\left(s_1\right), \ldots, G(s_{i-1}))$ for all $i \in [c(n)]$. We say that the construction is non-adaptive *if every S_i depends only on x and k.*

In our proof, we consider a restricted notion of black-box reductions that have access to a distinguisher with oracle access to a function. Roughly, this class captures reductions where the number of times it can query the adversary is bounded by a polynomial for all sufficiently good adversaries.

Definition 3.6 (Query-bounded reductions)**.** *Let (F, R) be a fully black-box construction of a PRF from an $r(n)$-bit stretch PRG, where*

$$F = \left\{ f_k^{(\cdot)} : \{0,1\}^{\mathsf{in}(n)} \to \{0,1\}^{\mathsf{out}(n)} \right\}_{n \in \mathbb{N}, k \in \{0,1\}^{\lambda(n)}},$$

let $\varepsilon : \mathbb{N} \to [0,1]$, and let $d : \mathbb{N} \to \mathbb{N}$. We say that $R^{G,\mathcal{A}}$ is (d, ε)-query-bounded *if for every oracle-aided algorithm $\mathcal{A}$ such that*

$$\left| \Pr_{k \leftarrow \{0,1\}^{\lambda(n)}} \left[\mathcal{A}^{f_k^G}(1^n) = 1 \right] - \Pr_{f \leftarrow \mathcal{F}_{\mathsf{in}(n), \mathsf{out}(n)}} \left[\mathcal{A}^{f}(1^n) = 1 \right] \right| \geq 1 - \varepsilon(n)$$

for infinitely many n's, $R^{G,\mathcal{A}}$ calls $\mathcal{A}$ at most $d(n)$ times.

Note that any reduction that is (d, ε)-query-bounded is also (d, ε')-query-bounded for any ε' such that $\varepsilon'(n) \geq \varepsilon(n)$. Therefore, for our impossibility result, proving it for a *larger* ε yields a stronger result.

4 Lower Bounds for Fully Black-Box Constructions of PRFs From PRGs

In this section, we prove our main result. We show that there is no fully black-box construction of PRF with input length in from a PRG with c calls to the PRG and using a query-bounded reduction, where $c = o(n/\log n)$ and $c = o(\mathsf{in}/\log \mathsf{in})$.

We are now ready to state and prove our main result.

Theorem 4.1. *Fix polynomials $r : \mathbb{N} \to \mathbb{N}$ and $\mathsf{in} : \mathbb{N} \to \mathbb{N}$ such that $\mathsf{in}(n) = \omega(\log n)$. Then for every $c : \mathbb{N} \to \mathbb{N}$ such that $c(n) = o\left(n/\log n\right)$ and $c(n) = o\left(\mathsf{in}/\log \mathsf{in}\right)$, and for every polynomial $d : \mathbb{N} \to \mathbb{N}$, there is no $c(n)$-call non-adaptive fully black-box construction of a weak PRF with input length $\mathsf{in}(n)$ and a single-bit output from an $r(n)$-bit stretch PRG, with a (d, e^{-n})-query-bounded reduction.*

Proof. We first give an overview of the proof. Our proof has 4 major steps.

Step 1: We fix m inputs $x_1, \ldots, x_m$ for a sufficiently large m. Then, for every key k, we identify a large subset of the inputs $\mathcal{I}_k$ based on repetitions in the seeds generated by the key and the inputs in $\mathcal{I}_k$. Specifically, we show that either there are c seeds that are common to all inputs in $\mathcal{I}_k$, or there $t < c$ common seeds and the rest of the seeds are mutually disjoint. Importantly, this step is independent of the PRG, i.e., the set $\mathcal{I}_k$ is the same for every PRG. We formalize the properties of the set $\mathcal{I}_k$ and show its existence in Lemma 4.3.

Step 2: Next, we define an "ideal adversary" $\mathcal{A}_{\mathsf{ideal}}^{f,G}$ that breaks the PRF by inverting the PRG G. This adversary is not optimal. Instead of outputting 'PRF' when all input-output pairs (given by the oracle f) are consistent with respect to some key k, we define a score function based on the number of consistent input-output pairs in the set $\mathcal{I}_k$, and let $\mathcal{A}_{\mathsf{ideal}}^{f,G}$ output 'PRF' if and only if the score is larger then a random threshold sampled uniformly at random. We prove, nonetheless, that this ideal adversary breaks the PRF. This is formalized in Claim 4.6.

Step 3: We then define a "real adversary" whose goal is to simulate the ideal adversary without inverting the PRG. It is given the query-answer pairs generated by the reduction's interaction with the PRG. Roughly, the difference between the real adversary and the ideal adversary is in the definition of the score functions they compute. We show that the real adversary behaves close to the ideal adversary. We formally prove this in Lemma 4.10.

Step 4: Finally, we combine the previous steps to argue that the reduction can successfully break the PRG using the real adversary. Since the real adversary does not query the PRG, we conclude the existence of an algorithm for breaking the PRG using a sub-exponential number of queries. Thus, instantiating the PRG with a random function will contradict Lemma 3.3, concluding the proof.

Assume towards contradiction that the claim is false. That is, there exists polynomials $\mathsf{in} = \mathsf{in}(n)$, $\lambda = \lambda(n)$, and $r = r(n)$, there exists $c = c(n)$ such that $c = \frac{n}{w(n)\log n}$ and $c = \frac{\mathsf{in}}{w(n)\log \mathsf{in}}$ for some $w(n) = \omega(1)$, there exists a polynomial $d = d(n)$, and there exists a fully black-box construction (F, R) such that R is (d, e^{-n})-query-bounded. We write

$$F = \left\{ f_k^{(\cdot)} : \{0,1\}^{\mathsf{in}} \to \{0,1\} \right\}_{n \in \mathbb{N}, k \in \{0,1\}^\lambda}$$

where $f_k^{(\cdot)}$ is given by

$$f_k^G(x) = P\left(x, k, G(S_1(x,k)), G(S_2(x,k)), \ldots, G(S_c(x,k))\right),$$

for efficiently computable functions $S_1, S_2, \ldots, S_c : \{0,1\}^{\mathsf{in}} \times \{0,1\}^\lambda \to \{0,1\}^n$ and $P : \{0,1\}^{\mathsf{in}} \times \{0,1\}^\lambda \times (\{0,1\}^{n+r})^c \to \{0,1\}$.

The proof follows the meta-reduction paradigm. We first define an "ideal" adversary that breaks the PRF using an *exponential number* of queries to the PRG. We then construct an *inefficient* "real" adversary with *no* access to the PRG that is able to simulate the ideal adversary, when provided with the collection of query–answer pairs generated by the reduction's interaction with the PRG. We then conclude that there exists an algorithm that breaks the security of the PRG using fewer than $2^{n/4}$ queries, by running the reduction and using the real adversary.

For the argument to follow through, we have to ensure that the reduction cannot distinguish between the ideal and real adversaries. We are unable to show that the statistical distance between the responses of the adversaries is negligible. Instead, we show that the distance is $1/\operatorname{poly}(n)$, for some polynomial $\operatorname{poly}(n)$ that depends on both the number of calls d made by the reduction to the adversary, and the number of queries $m(n)$ made by the adversary (to its oracle function). Thus, $\operatorname{poly}(n)$ must be sufficiently large, implying that $m(n)$ must be sufficiently large compared to d. Consequently, d must be fixed prior to $m(n)$; otherwise, the argument would suffer from a circular dependence among d and $m(n)$. This requirement is exactly the reason we assume the reduction to be query-bounded, as it ensures that d is determined independently of the adversary. We note that if one could construct real and ideal adversaries whose response distributions have negligible statistical distance, our assumption would no longer be necessary.

We first introduce some notation. For every $t \in \{0, 1, \ldots, c\}$, let $m_t = m_t(n)$ be a parameter (not necessarily a polynomial) to be defined by the analysis below. Further, let $m = m_0$ and fix m distinct inputs $x_1, \ldots, x_m \in \{0,1\}^{\mathsf{in}}$.[4]

Step 1: Identifying frequent seeds. In the first step of the proof, our goal is to identify for every key $k \in \{0,1\}^\lambda$ the set $\mathcal{I}_k$ of inputs with many common seeds. Towards defining $\mathcal{I}_k$, we use the following claim, which states that for every set of $t < c$ seeds $\mathcal{S}$, if every PRG seed outside of $\mathcal{S}$ appears a few times among a subset $\mathcal{I}_0 \subseteq [m]$ of the inputs, then for a large fraction of these inputs, the only common seeds belong to $\mathcal{S}$.

Claim 4.2. *Fix a number $p \in [m]$ and a key $k \in \{0,1\}^\lambda$. Further fix a set of inputs $\mathcal{I}_0 \subseteq [m]$ and a set of seeds $\mathcal{S} \subseteq \{0,1\}^n$ of size $|\mathcal{S}| = t \in \{0, 1, \ldots, c-1\}$, such that for every $s \in \mathcal{S}$ and $i \in \mathcal{I}_0$ there exists $j \in [c]$ such that $s = S_j(x_i, k)$. Assume that there is no $s \in \{0,1\}^n \setminus \mathcal{S}$ and $\mathcal{I} \subseteq \mathcal{I}_0$ of size p such that*

$$s \in \bigcap_{i \in \mathcal{I}} \{S_j(x_i, k) : j \in [c]\}.$$

Then there exists a set $\mathcal{I}' \subseteq \mathcal{I}_0$ of size

$$|\mathcal{I}'| \geq \frac{|\mathcal{I}_0|}{(c-t)(p-1)+1} \geq \frac{|\mathcal{I}_0|}{(c-t)p},$$

such that the sets of PRG queries generated by them (outside of $\mathcal{S}$) are mutually disjoint, i.e.,

$$S_j(x_i, k) \notin \{S_{j'}(x_{i'}, k) : j' \in [c], i' \in \mathcal{I}' \setminus \{i\}\} \setminus \mathcal{S},$$

for every $i \in \mathcal{I}'$ and $j \in [c]$.

Proof. Consider the graph $H = (\mathcal{I}_0, E)$ where $\{i, i'\} \in E$ for $i \neq i'$ if and only if x_i and $x_{i'}$ share a common PRG query outside of $\mathcal{S}$, i.e.,

$$(\{S_j(x_i, k) : j \in [c]\} \setminus \mathcal{S}) \cap (\{S_j(x_{i'}, k) : j \in [c]\} \setminus \mathcal{S}) \neq \emptyset.$$

Observe that the set $\mathcal{I}'$ we aim to show exists, corresponds to an independent set in H. By assumption, every PRG query $s \in \{0,1\}^n \setminus \mathcal{S}$ appears at most $p-1$ among the set of all seeds

[4] Note that if we obtain a contradiction for all distinct inputs, then the same holds for random inputs (up to adding the birthday bound).

in $\{S_j(x_i,k) : i \in \mathcal{I}_0, j \in [c]\}$. Thus, for every vertex $i \in \mathcal{I}_0$, every $j \in [c]$ such that $S_j(x_i,k) \notin \mathcal{S}$ contributes 1 to the degree of i. Since by assumption there are exactly t indexes $j \in [c]$ such that $S_j(x_i,k) \in \mathcal{S}$, the degree of each vertex in H is at most $(c-t)(p-1)$. Therefore, by greedily picking vertices for an independent set and removing their neighbors, we conclude that there exists an independent set of vertices of size at least $|\mathcal{I}_0|/((c-t)(p-1)+1)$. □

Recall that to construct the two adversaries, we define two *score functions* (one for each adversary) that aim to compute for how many inputs the reduction returned the correct output (with respect to some key). The two score functions are defined with respect to each key, via a procedure that depends only on the construction of the PRF (in particular, it is independent of the output of the PRG). This is important, since the score function does not require the real adversary to make any PRG queries.

At a high level, the procedure works in iterations, where in each iteration it looks for a seed s that appears for sufficiently many inputs (i.e., for many of the x_i's, at least one of the seeds needed to compute $f_k^G(x_i)$ is s). The process continues until either no s appear many times, or until the process finds c such seeds. In the former case, we can apply Claim 4.2 to argue there is a large set of inputs with mutually disjoint set of seeds, while in the latter case, we will later replace all calls to the PRG with fixed values. The following lemma summarizes this.

Lemma 4.3. *For every key $k \in \{0,1\}^\lambda$, there exists a number $\ell(k) \in \{0,1,\ldots,c\}$, there exist $\ell(k)$ distinct seeds $\mathbf{s}_k = (s_{k,1},\ldots,s_{k,\ell(k)}) \in (\{0,1\}^n)^{\ell(k)}$, and there exists a set $\mathcal{I}_k \subseteq [m]$ such that the following holds.*

1. *If $\ell(k) = c$, then $|\mathcal{I}_k| = m_c$ and $s_{k,t} \in \bigcap_{i\in\mathcal{I}_k}\{S_j(x_i,k) : j \in [c]\}$ for every $t \in [c]$.*
2. *Otherwise, if $\ell(k) < c$, then $|\mathcal{I}_k| = \frac{m_{\ell(k)}}{(c-\ell(k)+1)m_{\ell(k)+1}}$,*

$$S_j(x_i,k) \notin \{S_{j'}(x_{i'},k) : j' \in [c], i' \in \mathcal{I}_k \setminus \{i\}\},$$

for every $i \in \mathcal{I}_k$ and $j \in [c]$, and there exists a set $\mathcal{I} \subseteq [m]$ of size $|\mathcal{I}| = m_{\ell(k)}$ such that $\mathcal{I}_k \subseteq \mathcal{I}$ and $s_{k,t} \in \bigcap_{i\in\mathcal{I}}\{S_j(x_i,k) : j \in [c]\}$ for every $t \in [\ell(k)]$.

Proof. The proof follows from greedily looking for the next seed $s_{k,t}$ and the (sufficiently large) set of inputs generating it, until we found c such seeds, or it does not exist and we can apply Claim 4.2. Formally, the lemma follows from considering the following procedure:

1. Let $\mathcal{I}_{k,0} = [m]$ and $t = 1$.
2. While $t \le c$ do the following:
   - If there exists a seed $s \in \{0,1\}^n \setminus \{s_{k,1},\ldots,s_{k,t-1}\}$ and $\mathcal{I} \subseteq \mathcal{I}_{k,t-1}$ of size $|\mathcal{I}| = m_t$ such that $s \in \bigcap_{i\in\mathcal{I}}\{S_j(x_i,k) : j \in [c]\}$, then let $s_{k,t}$ and $\mathcal{I}_{k,t}$ denote the lexicographically smallest such seed s and set $\mathcal{I}$, and increment t by 1.
   - Otherwise, by Claim 4.2, there exists a set of inputs $\mathcal{I} \subseteq \mathcal{I}_{k,t-1}$ of size

$$|\mathcal{I}| = \frac{|\mathcal{I}_{k,t-1}|}{(c-t)\cdot m_t} = \frac{m_{t-1}}{(c-t)\cdot m_t}$$

   such that

$$S_j(x_i,k) \notin \{S_{j'}(x_{i'},k) : j' \in [c], i' \in \mathcal{I} \setminus \{i\}\},$$

for every $i \in \mathcal{I}$ and $j \in [c]$. Let $\ell(k) = t - 1$, and let $\mathcal{I}_k$ denote lexicographically smallest such set $\mathcal{I}$, and halt.

3. Let $\ell(k) = c$ and $\mathcal{I}_k = \mathcal{I}_{k,c}$.

□

It will be convenient to consider the value $\nu = \min_k |\mathcal{I}_k|$.

Step 2: Defining the ideal adversary. We next define the first score function, which we call the *ideal score function*. It depends on the entire PRG oracle and will only be used by the ideal adversary. Given a key and a sequence of possible outputs $\mathbf{z} \in (\{0,1\})^m$,[5] the score is defined by using Lemma 4.3 to find the seeds that appear many times. It then replaces the output of G on those seeds with fixed values, and returns the number of inputs in $\mathcal{I}_k$ that are consistent with $\mathbf{z}$.

In what follows, for every pair of sequences $\mathbf{s} = (s_1, \ldots, s_t) \in (\{0,1\}^n)^t$ and $\mathbf{y} = (y_1, \ldots, y_t) \in (\{0,1\}^{n+r})^t$, define

$$F_{\mathbf{s},\mathbf{y}} = \left\{ f^{(\cdot)}_{k,\mathbf{s},\mathbf{y}} : \{0,1\}^{\mathsf{in}} \to \{0,1\} \right\}_{n \in \mathbb{N}, k \in \{0,1\}^\lambda},$$

where $f^{(\cdot)}_{k,\mathbf{s},\mathbf{y}}$ is the same as $f^{(\cdot)}_k$, except that for oracle queries in $\mathbf{s}$, the oracle answer for the seed s_i is replaced with y_i, for every $i \in [t]$.

Definition 4.4 (Ideal score function)**.** *For every key $k \in \{0,1\}^\lambda$, every sequence of possible outputs $\mathbf{z} = (z_1, \ldots, z_m)$, and every function $G : \{0,1\}^n \to \{0,1\}^{n+r}$, define the ideal score of k to be*

$$\mathsf{is}^G_k(\mathbf{z}) = \max_{\mathbf{y} \in (\{0,1\}^{n+r})^{\ell(k)}} \left(\left| \left\{ i \in \mathcal{I}_k : f^G_{k,\mathbf{s}_k,\mathbf{y}}(x_i) = z_i \right\} \right| \right).$$

Observe that if $\ell(k) = c$, then $\mathsf{is}^G_k(\mathbf{z})$ is independent of G.

Next, we define the "ideal" adversary $\mathcal{A}^{f,G}_{\mathsf{ideal}}$ for the PRF. This adversary queries the PRG on *exponentially many* inputs. We will then show that the reduction can simulate this adversary efficiently, resulting in an efficient distinguisher for the PRG. Roughly, $\mathcal{A}^{f,G}_{\mathsf{ideal}}$ works as follows. First, it queries the oracle f on $x_1, \ldots, x_m$. The adversary outputs PRF if and only if there is a key whose ideal score is at least $\alpha \cdot |\mathcal{I}_k|$, where $\alpha \in [0.8, 1]$ *is sampled uniformly at random.*

Algorithm 4.5 ($\mathcal{A}_{\mathsf{ideal}}$)**.**
Input: The security parameter 1^n.
Oracle access: A function $f : \{0,1\}^{\mathsf{in}} \to \{0,1\}$ and a PRG $G : \{0,1\}^n \to \{0,1\}^{n+r}$.

1. *Query G on every input in $\{0,1\}^n$.*
2. *Query f on $x_1, \ldots, x_m$. Let $\mathbf{z} = (z_1, \ldots, z_m)$ denote the respective answers.*
3. *Sample $\alpha \in [0.8, 1]$ uniformly at random.*
4. *Output 1 if and only if there exists $k \in \{0,1\}^\lambda$ such that $\mathsf{is}^G_k(\mathbf{z}) \geq \alpha \cdot |\mathcal{I}_k|$.*

We first show that $\mathcal{A}^{f,G}_{\mathsf{ideal}}$ indeed distinguishes the PRF from a random function.

[5] The score also implicitly depends on the inputs $x_1, \ldots, x_m$ we fixed earlier.

Claim 4.6. *For all $G : \{0,1\}^n \to \{0,1\}^{n+r}$,*

$$\left|\Pr_{k\leftarrow\{0,1\}^\lambda}\left[\mathcal{A}_{\mathsf{ideal}}^{f_k^G,G}(1^n)=1\right]-\Pr_{f\leftarrow\mathcal{F}_{\mathsf{in}}}\left[\mathcal{A}_{\mathsf{ideal}}^{f,G}(1^n)=1\right]\right|\geq 1-e^{\lambda+c(n+r)-\nu/10}.$$

Proof. We first show that $\mathcal{A}_{\mathsf{ideal}}^{f_k^G,G}(1^n)$ always outputs 1. Indeed, observe that

$$\mathsf{is}_k^G(\mathbf{z})=\max_{\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)}}\left(\left|\left\{i\in\mathcal{I}_k : f_{k,\mathbf{s}_k,\mathbf{y}}^G(x_i)=f_k^G(x_i)\right\}\right|\right).$$

Then taking $y_i=G(s_{k,i})$ for every $i\in[\ell(k)]$ results in $\mathsf{is}_k^G(\mathbf{z})=|\mathcal{I}_k|$.

To conclude the proof, we next show that for a random function f, $\mathcal{A}_{\mathsf{ideal}}^{f,G}(1^n)$ outputs 1 with negligible probability. For every key $k\in\{0,1\}^\lambda$, every $\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)}$, and every $i\in\mathcal{I}_k$, define the random variable $X_{k,\mathbf{y},i}$ to be the indicator for the event $f_{k,\mathbf{s}_k,\mathbf{y}}^G(x_i)=f(x_i)$. Then

$$\mathsf{is}_k^G(\mathbf{z})=\max_{\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)}}\left(\sum_{i\in\mathcal{I}_k}X_{k,\mathbf{y},i}\right).$$

Since f is a random function, these indicators are independent and have expectation $1/2$. Therefore, for every $k\in\{0,1\}^\lambda$ and every $\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)}$, Hoeffding's inequality (see Theorem 3.1) implies that

$$\Pr_{f\leftarrow\mathcal{F}_{\mathsf{in}}}\left[\sum_{i\in\mathcal{I}_k}X_{k,\mathbf{y},i}>0.8\cdot|\mathcal{I}_k|\right]\leq e^{-2\cdot(0.8-0.5)^2\cdot|\mathcal{I}_k|}\leq e^{-|\mathcal{I}_k|/10}\leq e^{-\nu/10}.$$

Therefore, by the union bound,

$$\begin{aligned}\Pr_{f\leftarrow\mathcal{F}_{\mathsf{in}}}\left[\mathcal{A}_{\mathsf{ideal}}^{f,G}(1^n)=1\right]&\leq\Pr_{f\leftarrow\mathcal{F}_{\mathsf{in}}}\left[\exists k\in\{0,1\}^\lambda : \mathsf{is}_k^G(f(x_1),\ldots,f(x_m))\geq 0.8\cdot|\mathcal{I}_k|\right]\\&=\Pr_{f\leftarrow\mathcal{F}_{\mathsf{in}}}\left[\exists k\in\{0,1\}^\lambda\ \exists\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)} : \sum_{i\in\mathcal{I}_k}X_{k,\mathbf{y},i}\geq 0.8\cdot|\mathcal{I}_k|\right]\\&\leq 2^{\lambda+c(n+r)}\cdot e^{-\nu/10}\\&\leq e^{\lambda+c(n+r)-\nu/10}.\end{aligned}$$

□

Recall that for every $k\in\{0,1\}^\lambda$, $|\mathcal{I}_k|=m_c$ or $|\mathcal{I}_k|=\frac{m_{t-1}}{(c-t+1)m_t}$ for some $t\in[c]$. Since $\nu\geq|\mathcal{I}_k|$ for every $k\in\lambda$, taking $m_c\geq 10(\lambda+c(n+r)+n)$ and $m_{t-1}\geq(c-t+1)m_t\cdot 10(\lambda+c(n+r)+n)$ for all $t\in[c]$, results in the ideal adversary's advantage begin at least $1-e^{-n}$.

Step 3: Defining the real adversary. We now define the *real score*, which will be used by the real adversary. It is defined with respect to a set of query-answer pairs $\mathcal{G}$ instead of a PRG. The set $\mathcal{G}$ will later be instantiated with respect to the queries the reduction made to the PRG. Similarly to the ideal score, the computation of the real score also starts by replacing calls to G on seeds that appear many times with fixed inputs. The only difference is the end of the recursive process. There are two cases: either every call to G was replaced, or the inputs in $\mathcal{I}_k$ have mutually

disjoint seeds. In the former case, the real score is identical to the ideal score. In the latter case, the score is defined as follows. Every input x_i for which every remaining seed (i.e., for which the output of G was not fixed) belongs to the set $\mathcal{G}$ can be computed using G. Such inputs contribute the same value as in the ideal score. For other inputs, the real score estimates its contribution to the ideal score by computing the probability over the PRG that the answer is correct. Since this is the case where all remaining seeds are mutually disjoint, the outputs corresponding to these inputs are independent random variables, and we can estimate their contribution to the score with high precision (see Lemma 4.10 below).

In the following, for a set of PRG query-answer pairs $\mathcal{G} = \{(s_1^{\mathcal{G}}, y_1^{\mathcal{G}}), \ldots, (s_q^{\mathcal{G}}, y_q^{\mathcal{G}})\} \subseteq \{0,1\}^n \times \{0,1\}^{n+r}$ we let $\mathbf{s}^{\mathcal{G}}$ denote the sequence of queries in $\mathcal{G}$ that *do not appear in* $\mathbf{s}_k$, and let $\mathbf{y}^{\mathcal{G}}$ denote the corresponding sequence of outputs. The reason not to include the queries in $\mathbf{s}_k$ is that any query s made by the reduction is unknown to the ideal adversary, which maximizes its score over all possible values of $G(s)$. Thus, the real adversary does the same, obtaining consistency between the two scores.

Definition 4.7 (Real score function)**.** *For every key $k \in \{0,1\}^\lambda$, every sequence of possible outputs $\mathbf{z} = (z_1, \ldots, z_m)$, and every set of PRG query-answer pairs $\mathcal{G} = \{(s_1^{\mathcal{G}}, y_1^{\mathcal{G}}), \ldots, (s_q^{\mathcal{G}}, y_q^{\mathcal{G}})\} \in \{0,1\}^n \times \{0,1\}^{n+r}$, define $\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z})$ as follows.*

- *If $\ell(k) = c$, then let*

$$\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z}) = \max_{\mathbf{y} \in (\{0,1\}^{n+r})^c} \left(\left| \left\{ i \in \mathcal{I}_k : P\left(x_i, k, y'_1, \ldots, y'_c\right) = z_i \right\} \right| \right),$$

  where for every $j \in [c]$, $y'_j = y_{j'}$ for the unique $j' \in [c]$ such that $S_j(x_i, k) = s_{k,j'}$.

- *If $\ell(k) < c$, then let*

$$\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z}) = \max_{\mathbf{y} \in (\{0,1\}^{n+r})^{\ell(k)}} \left(\sum_{i \in \mathcal{I}_k} \Pr_{G' \leftarrow \mathcal{F}_{n,n+r}} \left[f^{G'}_{k,(\mathbf{s}_k || \mathbf{s}^{\mathcal{G}}),(\mathbf{y} || \mathbf{y}^{\mathcal{G}})}(x_i) = z_i \right] \right).$$

Similarly to the ideal score, if $\ell(k) = c$ then $\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z})$ is independent of $\mathcal{G}$. In particular, in such a case, $\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z}) = \mathsf{is}_k^G(\mathbf{z})$.

Since every ideal adversary we consider has advantage at least $1 - e^{-n}$, it follows that there exists a constant $\gamma \in \mathbb{N}$ such that for any $G \in \mathcal{F}_{n,n+r}$,

$$\left| \Pr_{s \leftarrow \{0,1\}^n} \left[R^{\mathcal{A}_{\mathsf{ideal}}, G}(1^n, G(s)) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^{n+r}} \left[R^{\mathcal{A}_{\mathsf{ideal}}, G}(1^n, y) = 1 \right] \right| \geq \frac{1}{n^\gamma} \tag{1}$$

for infinitely many n's. To obtain a contradiction, we construct an algorithm $\widetilde{R}^{(\cdot)}$ that successfully breaks a *randomly chosen* PRG with noticeable probability, using at most $2^{n/4}$ queries to G and *without* having access to $\mathcal{A}_{\mathsf{ideal}}$. Specifically, we define $\widetilde{R}$ such that

$$\Pr_{G \leftarrow \mathcal{F}_{n,n+r}} \left[\left| \Pr_{s \leftarrow \{0,1\}^n} \left[\widetilde{R}^G(1^n, G(s)) = 1 \right] - \Pr_{y \leftarrow \{0,1\}^{n+r}} \left[\widetilde{R}^G(1^n, y) = 1 \right] \right| \leq \frac{1}{3n^\gamma} \right] \leq \frac{3}{4}, \tag{2}$$

for infinitely many n's. Note that Equation (2) contradicts Lemma 3.3.

Before defining $\widetilde{R}$, we first introduce an adversary $\mathcal{A}_{\mathsf{real}}$ that, intuitively, efficiently emulates $\mathcal{A}_{\mathsf{ideal}}$ using the set of queries to G used so far by the reduction. Roughly, $\mathcal{A}_{\mathsf{real}}$ works similarly to

$\mathcal{A}_{\mathsf{ideal}}$, but uses rs instead of is. We then show by letting $\widetilde{R}$ run R, and replacing each call to $\mathcal{A}_{\mathsf{ideal}}$ with running $\mathcal{A}_{\mathsf{real}}$ using the set of query-answer pairs of the PRG used so far, the answers of the two distinguishing oracles are close in statistical distance.

Algorithm 4.8 ($\mathcal{A}_{\mathsf{real}}$)**.**
Input: The security parameter 1^n, a sequence $\mathbf{z} \in \{0,1\}^m$, and a set of query-answer pairs $\mathcal{G} \subseteq \{0,1\}^n \times \{0,1\}^{n+r}$.

1. *Sample $\alpha \in [0.8, 1]$ uniformly at random.*
2. *Output 1 if and only if there exists $k \in \{0,1\}^\lambda$ such that $\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z}) \geq \alpha \cdot |\mathcal{I}_k|$.*

We are now ready to define $\widetilde{R}$. The algorithm simply runs R, replacing all calls to $\mathcal{A}_{\mathsf{ideal}}$ by running $\mathcal{A}_{\mathsf{real}}$.

Algorithm 4.9 ($\widetilde{R}$)**.**
Input: The security parameter 1^n and a challenge $y^* \in \{0,1\}^{n+r}$.
Oracle access: A PRG $G : \{0,1\}^n \to \{0,1\}^{n+r}$.

1. *Simulate $R^{\mathcal{A},G}$ as follows.*
   - *Whenever R calls G, query G with the same input.*
   - *Whenever R calls $\mathcal{A}_{\mathsf{ideal}}$, send it $x_1, \ldots, x_m$ (if R queries G during this call then query it as well). Let $\mathbf{z} = (z_1, \ldots, z_m) \in \{0,1\}^m$ denote the respective answers of R. Call $\mathcal{A}_{\mathsf{real}}(1^n, \mathbf{z}, \mathcal{G})$, where $\mathcal{G}$ is the current set of query-answer pairs with respect to G, and send to R the same answer.*
2. *Output whatever R outputs.*

We next show that $\widetilde{R}$ and R output the same value with high probability. We do so by showing that the statistical distance between the states of R and $\widetilde{R}$ after every call to the adversary is small. Towards proving this, we first show that for every possible sequence of responses $\mathbf{z}$ made by the reduction, the two scores computed by the real and ideal adversaries at any given call are somewhat close except with a negligible probability, where the probability is taken over G and y^* (the randomness of the reduction is fixed). In the following, for a set $\mathcal{G} \subseteq \{0,1\}^n \times \{0,1\}^{n+r}$ we let $\mathcal{F}_{n,n+r}^{\mathcal{G}}$ denote the set of all functions that are consistent with $\mathcal{G}$.

Lemma 4.10. *Consider a call made by the reduction to the distinguishing oracle, and fix the query-answer pairs $\mathcal{G} \subseteq \{0,1\}^n \times \{0,1\}^{n+r}$ made so far by the reduction. Let Y denote the distribution over $\{0,1\}^{n+r}$ for the challenge given to the reduction (i.e., it is either uniform over $\{0,1\}^{n+r}$ or is $G(S)$ for S uniform over $\{0,1\}^n$), and let $\mathbf{z} \in (\{0,1\})^m$ denote the reduction's answers to the distinguishing oracle on inputs $x_1, \ldots, x_m$. Then for every $\delta \geq 2/\nu$,*

$$\Pr_{G \leftarrow \mathcal{F}_{n,n+r}^{\mathcal{G}}, y^* \leftarrow Y}\left[\exists k \in \{0,1\}^\lambda : \left|\mathsf{is}_k^G(\mathbf{z}) - \mathsf{rs}_k^{\mathcal{G}}(\mathbf{z})\right| \geq \delta \cdot |\mathcal{I}_k|\right] \leq e^{\lambda - \frac{\delta^2}{2} \cdot \nu}. \tag{3}$$

Proof. By the union bound, it suffices to show that

$$\Pr_{G\leftarrow\mathcal{F}^{\mathcal{G}}_{n,n+r},y^*\leftarrow Y}\left[\left|\mathsf{is}^G_k(\mathbf{z})-\mathsf{rs}^{\mathcal{G}}_k(\mathbf{z})\right|\geq\delta\cdot|\mathcal{I}_k|\right]\leq e^{-\frac{\delta^2}{2}\cdot\nu} \tag{4}$$

for every key $k\in\{0,1\}^\lambda$.

As noted before, if $\ell(k)=c$, then the two scores are identical and the difference is 0. In the remainder of the proof, we assume that $\ell(k)<c$.

For every $\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)}$, let

$$\mathsf{is}^G_k(\mathbf{z},\mathbf{y})=\left|\left\{i\in\mathcal{I}_k:f^G_{k,\mathbf{s}_k,\mathbf{y}}(x_i)=z_i\right\}\right|$$

and let

$$\mathsf{rs}^{\mathcal{G}}_k(\mathbf{z},\mathbf{y})=\sum_{i\in\mathcal{I}_k}\Pr_{G'\leftarrow\mathcal{F}_{n,n+r}}\left[f^{G'}_{k,(\mathbf{s}_k||\mathbf{s}^{\mathcal{G}}),(\mathbf{y}||\mathbf{y}^{\mathcal{G}})}(x_i)=z_i\right],$$

where $\mathbf{s}^{\mathcal{G}}$ is the sequence of seeds in $\mathcal{G}$ that do not appear in $\mathbf{s}_k$, and $\mathbf{y}^{\mathcal{G}}$ is the sequence of corresponding outputs.

We show that for any sequence $\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)}$,

$$\Pr_{G\leftarrow\mathcal{F}^{\mathcal{G}}_{n,n+r},y^*\leftarrow Y}\left[\left|\mathsf{is}^G_k(\mathbf{z},\mathbf{y})-\mathsf{rs}^{\mathcal{G}}_k(\mathbf{z},\mathbf{y})\right|\geq\delta\cdot|\mathcal{I}_k|\right]\leq e^{-\frac{\delta^2}{2}\cdot\nu}. \tag{5}$$

In particular, this holds for the $\mathbf{y}$ that maximizes the scores, which implies Equation (4). Note that the string $\mathbf{y}$ that maximizes the scores for the real and ideal adversaries could be different, but Equation (5) implies Equation (4) nevertheless.

Fixing k and $\mathbf{y}\in(\{0,1\}^{n+r})^{\ell(k)}$, we simplify notation by writing is and rs for the ideal and real scores, respectively. We next show that

$$\mathsf{rs}\leq\mathrm{E}_{G\leftarrow\mathcal{F}^{\mathcal{G}}_{n,n+r},y^*\leftarrow Y}[\mathsf{is}]\leq\mathsf{rs}+1. \tag{6}$$

By linearity of expectation, the contribution of every $i\in\mathcal{I}_k$ to $\mathrm{E}_G[\mathsf{is}]$ is

$$\Pr_{G\leftarrow\mathcal{F}^{\mathcal{G}}_{n,n+r},y^*\leftarrow Y}\left[f^G_{k,\mathbf{s}_k,\mathbf{y}}(x_i)=z_i\right]$$

(where z_i may depend on y^* and G).

If Y is the uniform distribution over $\{0,1\}^{n+r}$, then

$$\Pr_{G\leftarrow\mathcal{F}^{\mathcal{G}}_{n,n+r},y^*\leftarrow Y}\left[f^G_{k,\mathbf{s}_k,\mathbf{y}}(x_i)=z_i\right]=\Pr_{G'\leftarrow\mathcal{F}^{\mathcal{G}}_{n,n+r},y^*\leftarrow Y}\left[f^{G'}_{k,(\mathbf{s}_k||\mathbf{s}^{\mathcal{G}}),(\mathbf{y}||\mathbf{y}^{\mathcal{G}})}(x_i)=z_i\right]$$

for every $i\in\mathcal{I}_k$, hence $\mathrm{E}_{G,y^*}[\mathsf{is}]=\mathsf{rs}$. Otherwise, if the distribution of Y is $G(S)$ where S is uniform over $\{0,1\}^n$, then every $i\in\mathcal{I}_k$ such that $z_i=P(x_i,k,\mathbf{y}')$, where Y appears in $\mathbf{y}'$, contributes 1 to $\mathrm{E}_{G,y^*}[\mathsf{is}]$. Since the sets $\{S_j(x_i,k):j\in[c]\}$ are pairwise disjoint, there is at most 1 such i. On the other hand, for the real score, each i contributes a value between 0 and 1, thus $\mathrm{E}_{G,y^*}[\mathsf{is}]\leq\mathsf{rs}+1$.

For every $i\in\mathcal{I}_k$ let X_i denote the indicator for the event $f^G_{k,\mathbf{s}_k,\mathbf{y}}(x_i)=z_i$ (over sampling $G\leftarrow\mathcal{F}^{\mathcal{G}}_{n,n+r}$ and $y^*\leftarrow Y$). Then $\mathsf{is}=\sum_{i\in\mathcal{I}_k}X_i$. These random variables are independent since the

sets $\{S_j(x_i,k) : j \in [c]\}$ are pairwise disjoint. Since $\delta \geq 2/\nu \geq 2/|\mathcal{I}_k|$, by Hoeffding's inequality, it follows that

$$\begin{aligned}
\Pr_{G \leftarrow \mathcal{F}^{\mathcal{G}}_{n,n+r}, y^* \leftarrow Y}\left[|\mathsf{is} - \mathsf{rs}| \geq \delta \cdot |\mathcal{I}_k|\right] &\leq \Pr_{G \leftarrow \mathcal{F}^{\mathcal{G}}_{n,n+r}, y^* \leftarrow Y}\left[|\mathsf{is} - \mathrm{E}\left[\mathsf{is}\right]| \geq \delta \cdot |\mathcal{I}_k| - 1\right] \\
&\leq \Pr_{G \leftarrow \mathcal{F}^{\mathcal{G}}_{n,n+r}, y^* \leftarrow Y}\left[|\mathsf{is} - \mathrm{E}\left[\mathsf{is}\right]| \geq \frac{\delta}{2} \cdot |\mathcal{I}_k|\right] \\
&\leq e^{-\frac{2\left(\frac{\delta}{2} \cdot |\mathcal{I}_k|\right)^2}{|\mathcal{I}_k|}} \\
&= e^{-\frac{\delta^2}{2} \cdot |\mathcal{I}_k|} \\
&\leq e^{-\frac{\delta^2}{2} \cdot \nu}.
\end{aligned}$$

□

We now show that for any call by the reduction to the distinguishing oracle, the statistical distance (taken over G and the challenge y^*) between the answers of $\mathcal{A}_{\mathsf{ideal}}$ and $\mathcal{A}_{\mathsf{real}}$ is small.

Lemma 4.11. *Fix a call to the distinguishing oracle. Let $\mathcal{G} \subseteq \{0,1\}^n \times \{0,1\}^{n+r}$ denote the set of query-answer pairs made by R to G, let Y denote the distribution over the challenge, and let $\mathbf{z} = (z_1, \ldots, z_m) \in \{0,1\}^m$ denote the answers of R when receiving inputs $x_1, \ldots, x_m$ from the distinguishing oracle. Finally, for every $G \in \mathcal{F}^{\mathcal{G}}_{n,n+r}$ and $y^* \in \{0,1\}^{n+r}$ let ip^{G,y^*} denote the probability that $\mathcal{A}_{\mathsf{ideal}}(1^n)$ outputs 1 when given $\mathbf{z}$ as the oracle answers, and let $\mathsf{rp}^{\mathcal{G},y^*}$ denote the probability that $\mathcal{A}_{\mathsf{real}}(1^n, \mathbf{z}, \mathcal{G})$ outputs 1. Then, for every $\delta \geq 2/\nu$*

$$\mathrm{SD}_{G \leftarrow \mathcal{F}^{\mathcal{G}}_{n,n+r}, y^* \leftarrow Y}\left(\mathsf{ip}^{G,y^*}, \mathsf{rp}^{\mathcal{G},y^*}\right) \leq e^{\lambda - \frac{\delta^2}{2} \cdot \nu} + 5\delta.$$

Proof. For every $G \in \mathcal{F}^{\mathcal{G}}_{n,n+r}$ and every $y^* \in \{0,1\}^{n+r}$ let

$$\mu^{G,y^*} = \max_{k \in \{0,1\}^\lambda} \frac{\left|\mathsf{is}^G_k(\mathbf{z}) - \mathsf{rs}^{\mathcal{G}}_k(\mathbf{z})\right|}{|\mathcal{I}_k|}.$$

Additionally, for every $\alpha \in [0.8, 1]$ let

$$\mathsf{ip}^{G,y^*}(\alpha) = \begin{cases} 1 & \text{if } \exists k \in \{0,1\}^\lambda : \mathsf{is}^G_k(\mathbf{z}) \geq \alpha |\mathcal{I}_k| \\ 0 & \text{otherwise,} \end{cases}$$

and let

$$\mathsf{rp}^{\mathcal{G},y^*}(\alpha) = \begin{cases} 1 & \text{if } \exists k \in \{0,1\}^\lambda : \mathsf{rs}^{\mathcal{G}}_k(\mathbf{z}) \geq \alpha |\mathcal{I}_k| \\ 0 & \text{otherwise.} \end{cases}$$

Then by Jensen's inequality,

$$\left|\mathrm{E}_{\alpha \leftarrow [0.8,1]}\left[\mathsf{ip}^{G,y^*}(\alpha) - \mathsf{rp}^{\mathcal{G},y^*}(\alpha)\right]\right| \leq \mathrm{E}_{\alpha \leftarrow [0.8,1]}\left[\left|\mathsf{ip}^{G,y^*}(\alpha) - \mathsf{rp}^{\mathcal{G},y^*}(\alpha)\right|\right].$$

Note that any α contributes to the expectation on the right-hand side if and only if α is between $\mathsf{is}_k^G(\mathbf{z})/|\mathcal{I}_k|$ and $\mathsf{rs}_{k'}^{\mathcal{G}}(\mathbf{z})/|\mathcal{I}_{k',\ell'(k')}|$ for (possibly distinct) $k, k' \in \{0,1\}^\lambda$. Observe that this distance is at most μ^{G,y^*} (i.e., the distance with respect to the same key). Therefore,

$$\begin{aligned}
&\mathrm{E}_{\alpha\leftarrow[0.8,1]}\left[\left|\mathsf{ip}^{G,y^*}(\alpha)-\mathsf{rp}^{\mathcal{G},y^*}(\alpha)\right|\right]\\
&\quad\le \Pr_{\alpha\leftarrow[0.8,1]}\left[\exists k\in\{0,1\}^\lambda : \min\left\{\mathsf{is}_k^G(\mathbf{z}),\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z})\right\}\le\alpha\cdot|\mathcal{I}_k|\le\max\left\{\mathsf{is}_k^G(\mathbf{z}),\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z})\right\}\right]\\
&\quad\le \frac{\mu^{G,y^*}}{1-0.8}\\
&\quad= 5\mu^{G,y^*}.
\end{aligned}$$

Hence, by Lemma 4.10,

$$\begin{aligned}
&\Pr_{G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}},y^*\leftarrow Y}\left[\left|\mathrm{E}_\alpha\left[\mathsf{ip}^{G,y^*}(\alpha)-\mathsf{rp}^{\mathcal{G},y^*}(\alpha)\right]\right|\ge 5\delta\right]\\
&\quad\le \Pr_{G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}},y^*\leftarrow Y}\left[\exists k\in\{0,1\}^\lambda : \left|\mathsf{is}_k^G(\mathbf{z})-\mathsf{rs}_k^{\mathcal{G}}(\mathbf{z})\right|\ge\delta\cdot|\mathcal{I}_k|\right]\\
&\quad\le e^{\lambda-\frac{\delta^2}{2}\cdot\nu}.
\end{aligned}$$

Therefore, the statistical distance is bounded by

$$\begin{aligned}
\mathrm{SD}_{G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}},y^*\leftarrow Y}\left(\mathsf{ip}^{G,y^*},\mathsf{rp}^{\mathcal{G},y^*}\right) &\le \mathrm{E}_{G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}},y^*\leftarrow Y}\left[\left|\mathsf{ip}^{G,y^*}-\mathsf{rp}^{\mathcal{G},y^*}\right|\right]\\
&= \mathrm{E}_{G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}},y^*\leftarrow Y}\left[\left|\mathrm{E}_\alpha\left[\mathsf{ip}^{G,y^*}(\alpha)-\mathsf{rp}^{\mathcal{G},y^*}(\alpha)\right]\right|\right]\\
&\le \Pr_{G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}},y^*\leftarrow Y}\left[\left|\mathrm{E}_\alpha\left[\mathsf{ip}^{G,y^*}(\alpha)-\mathsf{rp}^{\mathcal{G},y^*}(\alpha)\right]\right|\ge 5\delta\right]+5\delta\\
&\le e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5\delta.
\end{aligned}$$

□

For every $t\in\{0,1,\ldots,d\}$, let st_t and $\widetilde{\mathsf{st}}_t$ denote the states of R and $\widetilde{R}$, respectively, after the tth call to the distinguishing oracle, where the state consists of, the challenge y^*, the (current) query-answer pairs $\mathcal{G}$ made to the PRG, and the t answers of the distinguishing oracle.[6] Then by Lemma 4.11 and the triangle inequality, for every $t\in[d]$, after the tth call to the distinguishing oracle,

$$\mathrm{SD}\left(\mathsf{st}_t,\widetilde{\mathsf{st}}_t\right)\le\mathrm{SD}\left(\mathsf{st}_{t-1},\widetilde{\mathsf{st}}_{t-1}\right)+e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5\delta\le t\cdot e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5t\delta,$$

where the statistical distances are taken over $G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}}$, $y^*\leftarrow Y$ where Y is the distribution of the challenge, and over the reductions' randomness. Thus,

$$\Pr\left[\widetilde{R}^G(y^*)\ne R^{\mathcal{A}_{\mathsf{ideal}},G}(y^*)\right]\le d\cdot e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5d\delta, \tag{7}$$

where the probability is taken over $G\leftarrow\mathcal{F}_{n,n+r}^{\mathcal{G}}$, $y^*\leftarrow Y$ where Y is the distribution of the challenge, and over the reductions' randomness.

[6] Formally, it should also depend on the randomness. Since all our claims hold for every randomness, we ignore it for simplicity.

Step 4: Putting it all together. Our goal is to upper-bound

$$\Delta := \Pr_{G\leftarrow\mathcal{F}_{n,n+r}}\left[\left|\Pr_{s\leftarrow\{0,1\}^n}\left[\widetilde{R}^G(1^n,G(s))=1\right]-\Pr_{y\leftarrow\{0,1\}^{n+r}}\left[\widetilde{R}^G(1^n,y)=1\right]\right|\leq\frac{1}{3n^\gamma}\right].$$

Towards bounding Δ, we first upper-bound the expected difference between the answers of R and $\widetilde{R}$ (where the expectation is taken over G). To alleviate notations, throughout the rest of the proof we remove $\mathcal{A}_{\mathsf{ideal}}$ from the notations and write R^G instead. Additionally, all probabilities are taken over the reductions' randomness and we ignore this for brevity.

Claim 4.12. *Let Y denote the distribution over the challenge. Then for all sufficiently large n's,*

$$\mathrm{E}_{G\leftarrow\mathcal{F}_{n,n+r}}\left[\left|\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=1\right]-\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right]\right|\right]\leq d\cdot e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5d\delta.$$

Proof. For every $G\in\mathcal{F}_{n,n+r}$, observe that

$$\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=1\right]\leq\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right]+\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)\neq R^G(y^*)\right]$$

and that

$$\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=1\right]\geq\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=R^G(y^*)\right]\cdot\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right].$$

Let $\widetilde{\mathcal{F}}$ denote the set of all functions in $\mathcal{F}_{n,n+r}$ such that

$$\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=1\right]\geq\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right].$$

Then

$$\begin{aligned}
&\mathrm{E}_{G\leftarrow\mathcal{F}_{n,n+r}}\left[\left|\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=1\right]-\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right]\right|\right]\\
&\quad\leq\frac{1}{2^{2n(n+r)}}\cdot\sum_{G\in\widetilde{\mathcal{F}}}\left(\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=1\right]-\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right]\right)\\
&\qquad+\frac{1}{2^{2n(n+r)}}\cdot\sum_{G\in\mathcal{F}_{n,n+r}\setminus\widetilde{\mathcal{F}}}\left(\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right]-\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)=1\right]\right)\\
&\quad\leq\frac{1}{2^{2n(n+r)}}\cdot\sum_{G\in\widetilde{\mathcal{F}}}\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)\neq R^G(y^*)\right]\\
&\qquad+\frac{1}{2^{2n(n+r)}}\cdot\sum_{G\in\mathcal{F}_{n,n+r}\setminus\widetilde{\mathcal{F}}}\Pr_{y^*\leftarrow Y}\left[R^G(y^*)=1\right]\cdot\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)\neq R^G(y^*)\right]\\
&\quad\leq\mathrm{E}_{G\leftarrow\mathcal{F}_{n,n+r}}\left[\Pr_{y^*\leftarrow Y}\left[\widetilde{R}^G(y^*)\neq R^G(y^*)\right]\right]\\
&\quad\leq d\cdot e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5d\delta,
\end{aligned}$$

where the last inequality follows from Equation (7). □

Now, observe that

$$\begin{aligned}\Delta \leq &\Pr_{G\leftarrow\mathcal{F}_{n,n+r}}\left[\left|\Pr_{s\leftarrow\{0,1\}^n}\left[R^G(1^n,G(s))=1\right]-\Pr_{y\leftarrow\{0,1\}^{n+r}}\left[R^G(1^n,y)=1\right]\right|<\frac{1}{n^\gamma}\right]\\ &+\Pr_{G\leftarrow\mathcal{F}_{n,n+r}}\left[\left|\Pr_{s\leftarrow\{0,1\}^n}\left[R^G(1^n,G(s))=1\right]-\Pr_{s\leftarrow\{0,1\}^n}\left[\widetilde{R}^G(1^n,G(s))=1\right]\right|>\frac{1}{3n^\gamma}\right]\\ &+\Pr_{G\leftarrow\mathcal{F}_{n,n+r}}\left[\left|\Pr_{y\leftarrow\{0,1\}^{n+r}}\left[R^G(1^n,y)=1\right]-\Pr_{y\leftarrow\{0,1\}^{n+r}}\left[\widetilde{R}^G(1^n,y)=1\right]\right|>\frac{1}{3n^\gamma}\right].\end{aligned}$$

Note that by Equation (1), the first term is 0 for infinitely many n's. As for the other two terms, by Markov's inequality and Claim 4.12, it follows that both terms are upper-bounded by

$$3n^\gamma\cdot\left(d\cdot e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5d\delta\right).$$

We conclude that

$$\Delta\leq 6n^\gamma\cdot\left(d\cdot e^{\lambda-\frac{\delta^2}{2}\cdot\nu}+5d\delta\right)$$

for infinitely many n's.

We next choose the parameters such that $\Delta\leq 3/4$ for infinitely many n's. Specifically, we choose δ and ν (which we later translate to choosing m_t for every $t\in[c]$). Let $\delta=1/48dn^\gamma\sqrt[4]{\nu}$ and let $\nu=2(\lambda+n)\cdot d^2n^{2\gamma}$. Then $\delta\geq 2/\nu$ and $\nu\geq 2(\lambda+n)/\delta^2$ if ν is sufficiently large. In this case, $e^{\lambda-\frac{\delta^2}{2}\cdot\nu}\leq e^{-n}$. Since ν is a polynomial of n, $\nu\leq 2de^n$ for all sufficiently large n's. Therefore $\delta\geq 2/\nu\geq e^{-n}/d$, hence $e^{-n}\leq d\delta$. Thus, for such choices of ν, $\Delta\leq 36dn^\gamma\cdot\delta\leq 3/4$.

Now, recall that $\nu=\min_{k\in\{0,1\}^\lambda}|\mathcal{I}_k|$, and that for every $k\in\{0,1\}^\lambda$, either $|\mathcal{I}_k|=m_c$ or $|\mathcal{I}_k|=\frac{m_{t-1}}{(c-t)m_t}$ for some $t\in[c]$. Further recall that for $\mathcal{A}_{\mathsf{ideal}}$ to be have $1-e^{-n}$ advantage, we took $m_c\geq 10(\lambda+c(n+r)+n)$ and $m_{t-1}\geq(c-t+1)m_t\cdot 10(\lambda+c(n+r)+n)$ for all $t\in[c]$. Therefore, we let[7]

$$m_c=10(\lambda+c(n+r)+n)+2(\lambda+n)\cdot d^2n^{2\gamma}$$

and

$$m_{t-1}=(c-t+1)\left(10(\lambda+c(n+r)+n)+2(\lambda+n)\cdot d^2n^{2\gamma}\right)\cdot m_t$$

for every $t\in[c]$. As a result, the adversaries query the function on $m=m_0\leq n^{\alpha c}\cdot c!$ inputs, for some constant $\alpha\in\mathbb{N}$.[8]. Since there are 2^{in} possible inputs, for the adversaries to be well-defined, it must be the case where $n^{\alpha c}\cdot c!\leq 2^{\mathsf{in}}$. Indeed, since $c=\frac{\mathsf{in}}{w(n)\cdot\log\mathsf{in}}$ for some $w(n)=\omega(1)$,

$$n^{\alpha c}\cdot c!\leq n^{\alpha c}\cdot c^c=2^{\alpha c\log n+c\log c}\leq 2^{\alpha\frac{\mathsf{in}}{w(n)\log\mathsf{in}}\log n+\frac{\mathsf{in}}{w(n)\log\mathsf{in}}\log\mathsf{in}}\leq 2^{\mathsf{in}},$$

for all sufficiently large n's, where the last inequality is due to the fact that $\mathsf{in}=\mathrm{poly}(n)$, hence $\log n=O(\log\mathsf{in})$.

Finally, in order to contradict Lemma 3.3, we show that the total number of queries $\widetilde{R}$ makes to G is at most $2^{n/4}$. Indeed, in each of the d calls to the distinguishing oracle, the number of calls to G that the reduction makes is at most

$$\mathrm{poly}(m)\leq n^{\beta c}\cdot(c!)^\beta\leq n^{\beta c}\cdot c^{\beta c}=2^{\beta c\log n+\beta c\log c}$$

[7]The bound on m_c can be improved to $10(\lambda+c(n+r)+n)$. This is because the other term comes from considering k's such that $\ell(k)<c$ (recall that if $\ell(k)=c$ then both scores are identical).

[8]α depends on the degree of the polynomials λ, r, d, and γ.

for some constant $\beta \in \mathbb{N}$. Recall that $c = \frac{n}{w(n)\cdot\log n}$, where $w(n) = \omega(1)$. Therefore, the total number of calls that $\widetilde{R}$ makes is at most

$$d \cdot 2^{\beta c \log n + \beta c \log c} = d \cdot 2^{\frac{\beta n}{w(n)} + \frac{\beta n}{w(n)\log n} \log \frac{n}{w(n)\log n}} \leq d \cdot 2^{\frac{2\beta n}{w(n)}} \leq 2^{n/4},$$

for sufficiently large n's. □

Remark 4.13 (On the definition of black-box reduction)**.** *In the definition we considered, the advantage of the reduction R is a polynomial of the advantage of the distinguishing oracle $\mathcal{A}$. This in particular allowed us to conclude the existence of $\gamma \in \mathbb{N}$ for which Equation (1) holds. In the definition for black box constructions of PRF from Beimel et al. [2], it is only required that if $\mathcal{A}$ succeeds with noticeable probability, then so does R. Under this definition, Equation (1) does not necessarily hold, since the advantage of R might decrease as a function of the advantage of $\mathcal{A}$.*

Nevertheless, to the best of our knowledge, all black-box reductions studied in the literature satisfy the stronger requirement. Moreover, our proof continues to hold under the weaker assumption that the advantage of R is bounded below for all distinguishing oracles with a sufficiently large advantage.

5 Lower Bounds for Black-Box Constructions of a PRF With a Long Output

In this section, we show our lower bounds for fully black-box constructions of PRF with a long output from a PRG, with adaptive calls. We will use the following result, stating the existence of a PRG with respect to some oracle f.

Lemma 5.1 ([2, Lemma 3.5])**.** *For every oracle $f = \{f_n : \{0,1\}^n \to \{0,1\}^n\}_{n\in\mathbb{N}}$ and $w, r : \mathbb{N} \to \mathbb{N}$ such that $n + r(n) = 2^{o(w(n))}$, there exists a (possibly inefficient) function $G = \left\{G_n : \{0,1\}^{w(n)} \to \{0,1\}^{n+r(n)}\right\}_{n\in\mathbb{N}}$ such that the following holds. For every* PPT *oracle-aided algorithm A, there exists a negligible function μ such that*

$$\left|\Pr_{s\leftarrow\{0,1\}^{w(n)}}\left[A^{f,G}\left(1^n, G\left(1^n, s\right)\right) = 1\right] - \Pr_{y\leftarrow\{0,1\}^{n+r(n)}}\left[A^{f,G}\left(1^n, y\right) = 1\right]\right| \leq \mu(n).$$

In what follows, we refer to the function G as the PRG with respect to the oracle f. We are now ready to state and prove our lower bound on the number of calls to the PRG required to construct a PRF with a long output. Throughout the section we fix the input length $\mathsf{in} = \mathsf{in}(n)$, the output length $\mathsf{out} = \mathsf{out}(n)$, the key length $\lambda = \lambda(n)$, and the PRG stretch $r = r(n)$.

Theorem 5.2. *For every $c(n) = \frac{\mathsf{out}}{r+\omega(\log n)}$, there is no $c(n)$-call fully black-box construction of a weak PRF with output length out from an r-bit stretch PRG.*

Proof. Fix $w = w(n) = \omega(\log n)$ and let $c = c(n) = \mathsf{out}/(r + w)$. Assume towards contradiction that there exists a c-call fully black-box construction (F, R) from an r-bit stretch PRG, where

$$F = \left\{f_k^{(\cdot)} : \{0,1\}^{\mathsf{in}} \to \{0,1\}^{\mathsf{out}}\right\}_{n\in\mathbb{N},k\in\{0,1\}^\lambda}.$$

That is, for every $i \in [c]$ there exist $S_i : \{0,1\}^{\mathsf{in}} \times \{0,1\}^\lambda \times (\{0,1\}^{(n+r)})^{i-1} \to \{0,1\}^n$, and there exists $P : \{0,1\}^m \times \{0,1\}^\lambda \times (\{0,1\}^{n+r})^c \to \{0,1\}^{\mathsf{out}}$ such that

$$f_k^G(x) = P\left(x, k, G\left(s_1\right), \ldots, G(s_c)\right)$$

for all $G : \{0,1\}^n \to \{0,1\}^{n+r}$, where $s_i = S_i(x, k, G(s_1), \ldots, G(s_{i-1}))$ for all $i \in [c]$.

To obtain a contradiction, we consider a PRG that is applied only to the first w bits of the seed. This, intuitively, implies that the entropy of the output of the PRF over m inputs is provided only by the key (which has λ entropy) and $m \cdot (w+r) \cdot c$ additional bits from all applications of the r-bit stretch PRG. On the other hand, for a random function, the amount of entropy is $m \cdot \mathsf{out}$. Taking $m = (n+\lambda)/c$ results in a large gap between the two cases.

In the following, for a string $s = (s_1, \ldots, s_n) \in \{0,1\}^n$ and $i \in [n]$ we let $s_{<i} = (s_1, \ldots, s_{i-1})$ and let $s_{\geq i} = (s_i, \ldots, s_n)$. We use $||$ to denote the concatenation of strings.

We first define a distinguisher $\mathcal{A}^{(\cdot)}$ for the PRF.

Algorithm 5.3 ($\mathcal{A}$)**.**
Input: The security parameter 1^n.
Oracle access: A function $f : \{0,1\}^{\mathsf{in}} \to \{0,1\}^{\mathsf{out}}$.

1. *Let* $m = m(n) = (n+\lambda)/c$ *and let* $x_1, \ldots, x_m \in \{0,1\}^{\mathsf{in}}$ *denote the lexicographically first* m *inputs.*[9]
2. *Query* f *on* $x_1, \ldots, x_m$*, and let* $z_1, \ldots, z_m \in \{0,1\}^{\mathsf{out}}$ *denote the respective answers.*
3. *Output 1 if and only if there exists* $k \in \{0,1\}^\lambda$ *and* $(y_{i,j})_{i\in[m],j\in[c]} \in (\{0,1\}^{w+r-1})^{m\cdot c}$ *such that*
$$P\left(x_i, k, (y_{i,1}||s_{i,1}), \ldots, (y_{i,c}||s_{i,c})\right) = z_i$$
for all $i \in [m]$*, where*
$$s_{i,j} = S_j\left(x_i, k, (y_{i,1}||s_{i,1}), \ldots, (y_{i,j-1}||s_{i,j-1})\right)_{\geq \frac{\mathsf{out}}{c}-r}$$
For all $i \in [m]$ *and* $j \in [c]$.

We next define a PRG for which $\mathcal{A}$ breaks the security of F. Since $w = \omega(\log n)$ and r is a polynomial, it follows that $n + r = 2^{o(w-1)}$. Thus, by Lemma 5.1 there exists a PRG $G' : \{0,1\}^{w-1} \to \{0,1\}^{w+r-1}$ with respect to the oracle $\mathcal{A}$, that cannot be broken with non-negligible probability. Consider the PRG $G : \{0,1\}^n \to \{0,1\}^{n+r}$ defined as
$$G(s) = G'(s_{<w})\,||s_{\geq w}.$$
Observe that $\mathcal{A}^{f_k^G}$ always outputs 1. To reach a contradiction, we next show that for a random function f, this probability is at most 2^{-n}. Indeed, by the union bound,

$$\begin{aligned}
&\Pr_{f\leftarrow\mathcal{F}_{\mathsf{in},\mathsf{out}}}\left[\mathcal{A}^f(1^n) = 1\right] \\
&= \Pr_{f\leftarrow\mathcal{F}_{\mathsf{in},\mathsf{out}}}\left[\exists k \in \{0,1\}^\lambda\ \exists (y_{i,j})_{i\in[m],j\in[c]} \in \left(\{0,1\}^{w+r}\right)^{m\cdot c}\ \forall i \in [m] : P\left(x_i, k, (y_{i,j}||s_{i,j})_{j=1}^c\right) = f(x_i)\right] \\
&\leq 2^{\lambda+m\cdot c\cdot(w+r-1)} \cdot \Pr_{f\leftarrow\mathcal{F}_{\mathsf{in}}}\left[\forall i \in [m] : P\left(x_i, k, (y_{i,j}||s_{i,j})_{j=1}^c\right) = f(x_i)\right] \\
&= \frac{2^{\lambda+m\cdot\mathsf{out}-m\cdot c}}{2^{m\cdot\mathsf{out}}} \\
&= 2^{-n}.
\end{aligned}$$

[9] Note that if we obtain a contradiction for all distinct inputs, then the same holds for random inputs, up to the birthday bound.

Since (F, R) is assumed to be a fully black-box PRF construction, it follows that $R^{\mathcal{A},G}$ breaks G. However, this implies the existence of a PPT algorithm $\widetilde{R}^{\mathcal{A},G'}$ breaking G', contradicting Lemma 5.1. □

Bibliography

[1] P. Baecher, C. Brzuska, and M. Fischlin. Notions of black-box reductions, revisited. In K. Sako and P. Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 296–315. Springer, 2013.

[2] A. Beimel, T. Malkin, and N. Mazor. Structural lower bounds on black-box constructions of pseudorandom functions. In L. Reyzin and D. Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part V*, volume 14924 of *Lecture Notes in Computer Science*, pages 459–488. Springer, 2024.

[3] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Proceedings of the 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques*, EUROCRYPT'06, page 409–426, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540345469.

[4] D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In K. Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 59–71. Springer, 1998.

[5] E. Bresson, J. Monnerat, and D. Vergnaud. Separation results on the "one-more" computational problems. In T. Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 71–87. Springer, 2008.

[6] M. Fischlin and D. Schröder. On the impossibility of three-move blind signature schemes. In H. Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 197–215. Springer, 2010.

[7] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, Redondo Beach, California, USA, November 12-14, 2000*, pages 305–313. IEEE Computer Society, 2000.

[8] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In D. S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 25–32. ACM, 1989.

[9] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

[10] I. Haitner, A. Rosen, and R. Shaltiel. On the (im)possibility of arthur-merlin witness hiding protocols. In O. Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 220–237. Springer, 2009.

[11] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In D. S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 44–61. ACM, 1989.

[12] L. A. Levin. One-way functions and pseudorandom generators. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 363–365, 1985.

[13] E. Miles and E. Viola. On the complexity of non-adaptively increasing the stretch of pseudorandom generators. In Y. Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pages 522–539. Springer, 2011.

[14] E. Miles and E. Viola. On the complexity of constructing pseudorandom functions (especially when they don't exist). *J. Cryptol.*, 28(3):509–532, 2015.

[15] R. Pass. Limits of provable security from standard assumptions. In L. Fortnow and S. P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 109–118. ACM, 2011.

[16] R. Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In A. Sahai, editor, *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, volume 7785 of *Lecture Notes in Computer Science*, pages 334–354. Springer, 2013.

[17] O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In M. Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.

[18] B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.