Lower Bounds on Constructions of Pseudorandom Generators from One-way Functions

Lower Bounds on Constructions of Pseudorandom Generators from One-way Functions

Makrand Sinha
makrand@student.ethz.ch

Master's Thesis

Advisor: Prof. Thomas Holenstein
thomas.holenstein@inf.ethz.ch

13 April 2011

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Complexity and Algorithms Research Group
Institute of Theoretical Computer Science
Department of Computer Science
ETH Zürich

Day after day, day after day,
We stuck, nor breath nor motion;
As idle as a painted ship
Upon a painted ocean.

Water, water, everywhere,
And all the boards did shrink;
Water, water, everywhere,
Nor any drop to drink.

The Rime of the Ancient Mariner
Samuel Taylor Coleridge

Abstract

Known constructions of pseudorandom generators from arbitrary one-way functions are quite inefficient. The best-known construction recently developed by Haitner, Reingold and Vadhan [14], when invoked on a one-way function with security parameter n, needs to query it on $\mathcal{O}(n^3 \log n)$ independent inputs, thus requiring a seed of length $\mathcal{O}(n^4 \log n)$ bits. However, at present no lower bounds on the efficiency of such constructions are known. It could very well be that one could construct a pseudorandom generator from any arbitrary one-way function with a single query.

In this thesis we make progress towards bridging this gap between upper and lower bounds on the efficiency of pseudorandom generators constructed from arbitrary one-way functions. Given any one-way function $f : \{0,1\}^n \to \{0,1\}^n$, we consider non-adaptive black-box constructions of pseudorandom generators of the form $g(x_1, \ldots, x_t, f(x_1), \ldots, f(x_t))$ for $t \geq 1$ and prove the following lower bounds on such constructions:

- For $t = 1$, we show that a black-box pseudorandom generator construction of the above form is impossible.
- For $t > 1$, we prove that if g satisfies a combinatorial condition (which depends on t), then $g(x_1, \ldots, x_t, f(x_1), \ldots, f(x_t))$ can not be a black-box pseudorandom generator construction. We conjecture that for $t \leq n/\log^2 n$, all constructions g of the form $g(x_1, \ldots, x_t, f(x_1), \ldots, f(x_t))$ satisfy this condition and under this conjecture we obtain a non-trivial $\Omega(n^2/\log^2 n)$ lower bound on the seed length of such pseudorandom generator constructions.

Acknowledgements

I have learned a lot about research during the last two year of my studies at ETH. When I first arrived here, I was unsure of myself and my abilities to do research. Today, however, I can gladly say that I have no such doubts.

A lot of people have impacted my thought process over the last couple of years. I was very fortunate that Thomas came back to ETH when I started my studies here. I had been thinking about constructions of pseudorandom generators from one-way functions for a while before coming here and this research would not have been possible if I had not found someone who was as enthusiastic about the problem as me. Most of this work has evolved over the last couple of years under his guidance and I am grateful for all the advice and encouragement he has provided to me over the time. I have learned a lot from him about research in general. I have seen him ask the right questions and formalize my incomprehensible ideas so easily. I have seen him be extremely patient and enjoy the false successes we had (in addition to the real ones). I hope that some of his positive attitude has rubbed off on me.

Konstantinos Panagiotou also deserves a special mention for being my first mentor and co-author and a great source of advice in my early years as a researcher. I am also greatly indebted to Angelika Steger for all the support and frank advice she has given me. If it wouldn't be for her, I would not be here at ETH.

I have also had the great pleasure of working with Uli Wagner and Jirka Matousek. I thank them for providing me constant encouragement to continue asking my dumb questions.

I would also like thank other students and staff here who have directly or indirectly helped me in one way or the other - Nicla Bernasconi, Dominik Scheder, Robin Moser, Robin Künzler, Chandan Dubey, Divesh Aggarwal, Ueli Peter. Andrea Salow and Floris Tschurr have helped me numerous times in my administrative and life troubles here in Zürich.

This work was supported by the Excellence Scholarship and Opportunity Programme of the ETH Zürich Foundation. I thank them for their generosity.

Special thanks to my friends - Cleonela Serban and Michael Belfrage. I owe Cleo for some great times and for sharing the study burden with me on mandatory courses which I vehemently did not want to take. To Mike I owe several awesome conversations about maths, theory, research and life in general.

I am deeply thankful to my parents and my brother for always supporting me, giving me so much freedom and for being proud of me.

I dedicate this thesis to the memory of my friend Shiva Khandelwal and my grandmother Shanti Sinha. They were and always will be a great source of inspiration and strength to me. The last word of acknowledgement is saved for Garima, the one person who means most to me. Without her support this tumultuous struggle would have been incredibly difficult.

Contents

In loving memory of my friend Shiva Khandelwal and my grandmother Shanti Sinha.

CHAPTER 1

Introduction

This thesis is concerned with lower bounds on the efficiency of pseudorandom generators constructed from one-way functions. A Pseudorandom Generator (PRG) is a deterministic algorithm which stretches a short random seed into a longer string which "appears" to be random to all polynomial time algorithms. The study of pseudorandom generators is relevant to modern computing for one very important reason.

Randomness is an essential resource in computer science today and plays key roles in many fundamental computational tasks like cryptography, distributed computing and many others. In fact, in many cases, randomized algorithms and protocols are simpler and more efficient as compared to their deterministic analogues. However, the model of randomness used in computer science assumes that we have an access to a long string of uniform random bits.

In reality, however, such strings are hard to come by. Even though it is widely believed that randomness exists in nature, it seems unclear what could be the source of a long string of pure distilled randomness. In practice today, most computational tasks that require access to randomness generate "random" bits in some ad-hoc way and it is generally not guaranteed that the provable guarantees which hold for algorithms using uniform random bits hold in these cases. For example, many cryptographic protocols are based on the assumption that it is possible to sample a long string of uniform random bits as needed. But if the string is not uniformly random, an adversary might be able to break the security of the protocol.

Pseudorandom generators play a very important part in closing the gap between the models of randomness in computation and what is available in reality. In many cases, one can assume access to short uniformly random strings (*e.g.* by employing a randomness extractor [see 1, Chapter 21] to a weak random source) and then stretch it to a longer pseudorandom string under which the theoretical guarantees of randomized algorithms and protocols do not deviate much from the uniform random setting.

Apart from the their connection to practice, pseudorandom generators have also proven to be of fundamental importance in the theory of computation itself. In cryptography, they are the basis of constructing other fundamental cryptographic primitives such as pseudorandom functions [8], bit-commitment schemes [20], and many more ([see 6]). In complexity, they are needed for example in the natural proofs barrier for circuit lower bounds ([21]).

However, it is not easy to come up with a natural candidate for a PRG as to unconditionally prove that such a generator exists is at least as difficult as proving $\mathsf{P} \neq \mathsf{NP}$. So, instead one seeks to construct PRGs from other basic primitives, like one-way functions, where such natural examples seem to be easily available. Assuming structural properties on one-way functions (*e.g.* considering one-way permutations) one can generate a pseudorandom string with only a seed of length $\mathcal{O}(n)$ where n is the security parameter for the original one-way function. However, under the assumption of one-wayness only, such constructions are quite inefficient: the best known construction recently discovered by Haitner, Reingold and Vadhan [14] requires seed length $\mathcal{O}(n^4 \log n)$. Note that this means that if the one-way function is secure only when applied on at least 100 random bits, then the constructed PRG gives a pseudorandom string only when roughly 10^8 random bits are used. It seems far-fetched to assume that such an inordinate number of pure random bits can be sampled in practice.

Considering the above, a fundamental question arises : are there some inherent limitations due to which the above constructions are inefficient? That is, can one even hope to give much more efficient construction, say one with $\mathcal{O}(n \log n)$ seed length, of PRGs from arbitrary one-way functions. The answer to this question involves proving lower bounds on the seed lengths required by such constructions. Currently however, no such lower bound is known. Thus, we cannot exclude the possibility that a single invocation of the one-way function is sufficient in order to get a pseudorandom generator. In such a case, the construction would be by far efficient enough to be used in practice. In this thesis we make progress towards resolving this unsatisfactory state of affairs by giving lower bounds on a certain class of constructions which are as strong as possible.

In the following section, we first give a brief history of constructions of PRGs from one-way functions. Section 1.2 gives a brief background on black-box constructions and techniques for proving impossibility results for such constructions. Finally, section 1.3 presents our results on lower bounds for a class of PRG constructions.

1.1 Constructions of PRGs from one-way functions

The roots of modern pseudorandom generators goes back to a paper by Blum and Micali [3]. Motivated by cryptographic applications, they considered the notion of a generator which on a random seed produced sequences of bits $b_1, \ldots, b_k$ in time polynomial in k such that the sequence of bits is unpredictable by probabilistic polynomial time (PPT) algorithms, *i.e.* no PPT algorithm can predict b_i from $b_1, \ldots, b_{i-1}$ with probability slightly more than $1/2$. Based on the hardness of discrete logarithm, they also gave such a generator. Yao [24] introduced the modern definition of pseudorandom generators and proved that it is equivalent to the one used in [3]. Furthermore, he also gave a much more general construction of such pseudorandom generators from any one-way permutation. (Later Goldreich and Levin [7] gave a much simpler construction of pseudorandom generators from any one-way permutation.)

At this point, people continued to search what minimal assumptions were needed to construct a pseudorandom generator. Levin [19] showed that the necessary and sufficient condition for the existence of a PRG was a one-way function which still remains one-way when iterated on itself. Later Goldreich, Krawczyk and Luby [9] used the result of [19] and showed how to construct a PRG from

a *regular* one way function (functions where each value in the range has roughly the same number of preimages).

A breakthrough came in a couple of papers by Impagliazzo, Levin and Luby [18] and Håstad [15]: together it showed that pseudorandom generators can be constructed from an arbitrary one-way function with no structural guarantees. In other words, pseudorandom generators exist if and only if one-way functions exist. The papers were merged into a single journal paper [10] and the construction is widely known as the HILL construction. This work introduced many new ideas which have proved very influential in the theory of computation.

Though the HILL construction takes polynomial time, it is highly inefficient and impractical as it requires a seed of length $\mathcal{O}(n^8)$ [1] where n is the security parameter of the original one-way function. Future directions of research focused upon giving a better construction for certain classes of functions. Holenstein [17] considered the constructions of pseudorandom generators from exponentially hard one-way functions (*i.e* no PPT algorithm can invert the one-way function with probability more than $2^{-\Omega(n)}$) and proved that the HILL construction in this case requires a seed of length $\mathcal{O}(n^5)$.

Haitner, Harnik and Reingold [13] revisited the technique used in [9] and used it to give an improved construction of PRG from general one-way functions with seed length $\mathcal{O}(n^7)$. They also give much efficient constructions of generators from regular one-way functions (seed length $\mathcal{O}(n \log n)$) and one-way functions with exponential hardness (seed length $\mathcal{O}(n^2)$).

Recently, Haitner, Reingold and Vadhan [14] gave a much simpler and efficient construction of pseudorandom generators from one-way functions. Not only their construction achieves a seed length of $\mathcal{O}(n^4 \log n)$ for general one-way functions, it is also non-adaptive and efficiently parallelizable. However, the seed length of $\mathcal{O}(n^4 \log n)$ for general one-way functions is still much worse as compared to the case of one-way permutation (where $\mathcal{O}(n)$ is possible due to [7]) which leads us to the main topic of this thesis - lower bounds on pseudorandom generator constructions.

1.2 Black Box Separation

Before we state the main results we prove, let us discuss the main techniques which we use. In this thesis, we are concerned with black-box constructions of pseudorandom generators from one-way functions. Informally, a construction of a primitive P from another primitive Q is termed *black-box* if it ignores the internal structure of Q's implementation and also has a black-box proof of security. That is, the adversary for breaking Q ignores the internal structure of both Q's implementation and of the (alleged) adversary breaking P. All known constructions of pseudorandom generators are black-box ones.

An important fact about black-box constructions is that they relativize, *i.e.* their security holds even in the presence of oracles. Thus, for proving that certain black-box constructions of PRGs from one-way functions is impossible, it suffices to give an oracle separating them, *i.e.* given access to the oracle an adversary can tell whether the output is pseudorandom or completely random while there exists a one-way function even in the presence of the oracle. Such an oracle is usually called *separation oracle*. The techniques for answering the latter (proving that one-way functions exist relative to the

[1] Håstad *et.al.* [10] presents a construction with seed length $\mathcal{O}(n^{10})$ and mention without proof that $\mathcal{O}(n^8)$ is the best possible that could be achieved using their ideas. Holenstein [17] gave a generalized proof for the HILL construction and formally proved the $\mathcal{O}(n^8)$ seed length.

oracle) seem to be quite limited and in most cases specific to the case of one-way permutations. However, many of them build upon the information-theoretic method of Gennaro and Trevisan [5]. Here one shows that if a polynomial-time adversary with access to the separation oracle Breaker inverts a random permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ on a non-negligible fraction of inputs, then one can store these inputs with very few bits. This implies that π can be represented in $o(\log(2^n!))$ bits and one can use this to argue that there exist one-way permutations even in the presence of the separation oracle.

Another technique which seems a bit easier to apply for proving the hardness of one-way functions relative to an oracle was developed by Haitner and Holenstein [12] following the work of Simon [23]. Informally, to show that an adversary with oracle access to Breaker and a one-way permutation π on $\{0,1\}^n$ can not find $\pi^{-1}(y)$ for $y \in \{0,1\}^n$ drawn uniformly at random, one modifies the permutation π by randomly selecting an $x \in \{0,1\}^n$ and mapping it to y. This new function is almost certainly not a permutation, but if one can show that with high probability the execution of the adversary remains the same even after the modification, then it implies that the adversary could not have found x as a pre-image. Since after the change both $\pi^{-1}(y)$ and x are equally likely to be found, the adversary could not have inverted y in the beginning with non-negligible probability.

To apply these techniques to prove lower bounds on PRG constructions, we extend the above techniques to general one-way functions (since due to [7], PRGs can be constructed from any one-way permutation with a single query, see Theorem 2.3.6 for details).

1.3 Lower Bounds - Results of This Thesis

Let us now be more precise about the results proven in this thesis. Taking cues from the non-adaptive black-box PRG construction of Haitner, Reingold and Vadhan [14], we consider non-adaptive pseudorandom generator construction of the form $g(x_1,\ldots,x_t,f(x_1),\ldots,f(x_t))$ for $t \geq 1$ where $f : \{0,1\}^n \to \{0,1\}^n$ is the one-way function. The main results of this thesis concern lower bounds on black-box constructions of the above form. (Note that proving lower bounds for non-black-box constructions is at least as difficult as proving $\mathsf{P} \neq \mathsf{NP}$.) The first main result is the following.

1.3.1 Theorem. (Black-box impossibility for PRGs with single query - Informal)
There exists no black-box pseudorandom generator construction of the form $g(x,f(x))$ where $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function.

To prove the above theorem, we first define a suitable separation oracle which breaks the security of g and then we extend the techniques discussed in the last section to show that there exist one-way functions which remain hard relative to this oracle. The key idea behind defining the separation oracle is to analyze g and based on that select a suitable family of hard functions $\mathscr{F}$ such that for some $f \in \mathscr{F}$, $g(x,f(x))$ is not secure while f is secure even in the presence of the oracle. In the case of $t = 1$, the selection of such a family $\mathscr{F}$ boils down to a combinatorial condition which is shown to be true by a simple argument.

For the cases where $t > 2$, to show a similar impossibility result we need a generalization of the above combinatorial condition (which depends on t). Unfortunately in these cases, the condition turns out to be quite complex and we could not prove that it is satisfied for every g of the above form. However, there are also a lot of other subtleties and technical issues in the black-box impossibility

proof for the cases with $t \geq 2$ which we show how to resolve. So, essentially proving black-box impossibility results on PRGs of the above form with $t \geq 2$, reduces down to proving that a certain combinatorial condition holds for all g of the above form. We believe that this is true for values of $2 \leq t \leq \frac{n}{\log^2 n}$ and under this conjecture we obtain non-trivial lower bounds of $\Omega(n^2/\log^2 n)$ on the seed length of black-box non-adaptive PRG constructions of the above form. The conjecture is a bit technical to state here, so we defer the statements of the conjecture to Chapter 5 and just state the main results for $t \geq 2$ here.

1.3.2 Theorem. (Black-box impossibility for PRGs with $2 \leq t \leq n/\log^2 n$ queries - Informal) *Assuming Conjecture 5.3.2, for every $2 \leq t \leq n/\log^2 n$, there exists no black-box pseudorandom generator construction of the form $g(x_1, \ldots, x_t, f(x_1), \ldots, f(x_t))$ where $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way function. In other words, there exists no black-box PRG construction of the form $g(x_1, \ldots, x_t, f(x_1), \ldots, f(x_t))$ from one-way functions $f : \{0,1\}^n \to \{0,1\}^n$ with seed length less than $n^2/\log^2 n$.*

1.4 Structure of This Thesis

In Chapter 2, we give some basic definitions and results from previous works that we need later. In Chapter 3 we review the construction given by Haitner, Reingold and Vadhan [14]. Though this is not particularly relevant for lower bounds except for some intuition, we feel that it is worthwhile to know about the best upper bounds before starting to work on lower bounds. In Chapter 4, we discuss in more detail what black-box separation means and illustrate the standard techniques to prove a simple impossibility result. Finally, Chapter 5 of this thesis contains our main results on lower bounds for PRG constructions.

CHAPTER 2

Definitions and Preliminaries

This chapter presents the basic notations, definitions and tools needed for this thesis. Most of these are pretty standard and can be found in standard texts on foundations of cryptography (c.f. [6]).

2.1 Notations

Let x and y be two bit strings. We denote by (x, y) the concatenation of x and y. If $x \in \{0,1\}^n$ is a n bit string, then $x|_{\{i\}}$ is the i^{th} bit of x, $x|_{\{i,\ldots j\}}$ is $(x|_{\{i\}}, \ldots, x|_{\{j\}})$. To make the notation simpler, we will sometimes use $(x_i, \ldots, x_j)$ to denote $x|_{\{i,\ldots j\}}$ when there are no other subscripts in use.

An $m \times n$ bit matrix x is indicated by $x \in \{0,1\}^{m\times n}$ and $x_{i,j}$ refers to the (i,j)-entry in x. We will sometimes view x as a sequence $x = x_1, \ldots, x_m$ of m strings, each of length n. The operation $\odot$ indicates matrix multiplication over GF[2]. Furthermore, for a function $f : \{0,1\}^n \to \{0,1\}^m$, the notation $f^{\otimes t}(x^{\otimes t}) := (f(x^{(1)}), \ldots, f(x^{(t)}))$ where $x^{\otimes t} = (x^{(1)}, \ldots, x^{(t)}), x^{(i)} \in \{0,1\}^n$.

We use the notation $\text{poly}(n)$ and $\text{poly}^{-1}(n)$ to denote the set of all positive polynomial functions and inverse polynomial functions respectively. Moreover, $\text{neg}(\text{n})$ is used to denote the class of negligible functions, *i.e.* functions which are smaller than any fixed positive polynomial $p(n)$ for large enough n.

2.2 Information Theoretic Notions

2.2.1 Distributions and Entropy

We denote by $X \sim_{\mathcal{D}} \{0,1\}^n$ that the random variable X is distributed over $\{0,1\}^n$ with the distribution $\mathcal{D}$. We will also sometimes abuse the same notation for a random sample drawn from a distribution, *i.e.* $x \sim_{\mathcal{D}} S$ will denote a random sample x which is drawn from set S according to the distribution $\mathcal{D}$. For a random variable distributed uniformly over a set S, we will write $X \sim_{\mathcal{U}} S$.

We let $\mathcal{U}_n$ denote the uniform distribution over $\{0,1\}^n$. As a matter of notational shorthand we will write $X \sim \mathcal{U}_n$ to mean that $X \sim_{\mathcal{U}} \{0,1\}^n$. Given a function $f : \{0,1\}^n \to \{0,1\}^m$ and a distribution $\mathcal{D}$, we denote by $f(\mathcal{D})$ the distribution induced on $\{0,1\}^m$ by f operating on the distribution $\mathcal{D}$.

The support of a random variable X is defined as $\mathsf{Supp}(X) = \{x : \mathbb{P}[X = x] > 0\}$. Let $\mathcal{D}$ be a distribution over some finite domain $\mathscr{U}$ and $X \sim \mathcal{D}$, then we define the following measure of entropy:

- For $x \in \mathsf{Supp}(X)$, we define the sample entropy of x to be $H_X(x) := \log(1/\mathbb{P}[X = x])$.
- The Shannon entropy of X is $H(X) = \mathbb{E}_{x \sim_{\mathcal{U}} X}[H_X(x)]$.
- The min-entropy of X is $H_\infty(X) = \min_{x \in \mathsf{Supp}(X)} H_X(x)$.
- The max-entropy of X is $H_0(X) = |\mathsf{Supp}(X)|$.

2.2.2 Statistical Distance

2.2.1 Definition (Statistical Distance)
Let $\mathcal{D}$ and $\mathcal{E}$ be distributions on a set X. The statistical distance between $\mathcal{D}$ and $\mathcal{E}$ is

$$\|\mathcal{D} - \mathcal{E}\| = \frac{1}{2} \sum_{x \in X} \Big| \mathbb{P}(\mathcal{D}(x)) - \mathbb{P}(\mathcal{E}(x)) \Big|.$$

We say two distributions are ε-close if the statistical distance between them is at most ε.

2.2.3 Universal Hash Functions and the Left Over Hash Lemma

2.2.2 Definition (2-Universal Hash Functions)
For each $n \in \mathbb{N}$, let $\mathscr{Q}_n$ be a family of functions mapping $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$. We say that the ensemble $\{\mathscr{Q}_n\}_{n \geq 1}$ is an efficient family of 2-universal hash functions if it is efficiently computable and for every distinct $x_1, x_2 \in \{0,1\}^n$ and every $y_1, y_2 \in \{0,1\}^{\ell(n)}$ the following holds

$$\mathbb{P}_{q \sim_{\mathcal{U}} \mathscr{Q}}[h(x_1) = y_1 \wedge h(x_2) = y_2] = 2^{-2\ell(n)}.$$

Note that the following is a 2-universal hash function family (c.f. [4]) : for a bitstring $y \in \ell(n)$, $q(x) = (x, 1) \odot y$. The above function family has description length at most $\ell(n)$.

Another useful family of 2-universal hash functions is the collection of $\ell \times n$ matrices with entries in $GF(2)$.

One of the most useful facts about universal hash functions (at least in the context of PRGs) is the following lemma.

2.2.3 Lemma. (Left over hash lemma [10], [17])
Let $\mathcal{D}$ be a probability distribution over $\{0,1\}^n$ with Min-entropy at least m and let κ be a positive integer. Let $\mathscr{Q}$ be a family of 2-universal hash function from $\{0,1\}^n$ to $\{0,1\}^{m-2\kappa}$. Let $X \sim \mathcal{D}$, $Q \sim_{\mathcal{U}} \mathscr{Q}$. Then

$$\|(Q(X), Q) - (U_{m-2\kappa}, Q)\| \leq 2^{-\kappa}$$

2.2.4 Flattening Shannon Entropy

The following is a well known result that Shannon entropy can be converted into Min-entropy by taking a direct product of random variables.

2.2.4 Lemma. ([14])

1. *Let X be a random variable taking values in a universe $\mathscr{U}$, let $t \in N$, and let $\varepsilon > 0$. Then with probability at least $1 - \varepsilon - 2^{-\Omega(t)}$ over $x \sim_{\mathcal{U}} X^{\otimes t}$,*

$$\|H_{X^{\otimes t}}(x) - t \cdot H(X)\| \leq \mathcal{O}(\sqrt{t \log(1/\varepsilon)} \cdot \log(|\mathscr{U}| \cdot t)).$$

2. *Let X, Y be jointly distributed random variables where X takes values in a universe $\mathscr{U}$, let $t \in N$, and let $\varepsilon > 0$. Then with probability at least $1 - \varepsilon - 2^{-\Omega(t)}$ over $(x, y) \sim_{\mathcal{U}} (X^{\otimes t}, Y^{\otimes t}) := (X, Y)^{\otimes t}$,*

$$\|H_{X^{\otimes t}|Y^{\otimes t}}(x|y) - t \cdot H(X|Y)\| \leq \mathcal{O}(\sqrt{t \log(1/\varepsilon)} \cdot \log(|\mathscr{U}| \cdot t)).$$

2.3 Cryptographic Primitives and Tools

2.3.1 Computational Indistinguishability

We define a *Distribution Ensemble* as a series $\{\mathcal{D}_n\}_{n \geq 1}$ where $\mathcal{D}_n$ is a distribution over $\{0,1\}^n$. Let $\{\mathcal{X}_n\}$ and $\{\mathcal{Y}_n\}$ be distribution ensembles. Define the distinguishing advantage of an algorithm A between the ensembles $\{\mathcal{X}_n\}$ and $\{\mathcal{Y}_n\}$ denoted as $\Delta_A(\{\mathcal{X}_n\}, \{\mathcal{Y}_n\})$, by:

$$\Delta_A(\{\mathcal{X}_n\}, \{\mathcal{Y}_n\}) = \left| \mathbb{P}[A(1^n, \mathcal{X}_n) = 1] - \mathbb{P}[A(1^n, \mathcal{Y}_n) = 1] \right|$$

where the probabilities are taken over the distributions $\mathcal{X}_n$ and $\mathcal{Y}_n$, and the randomness of A. We say that $\{\mathcal{X}_n\}$ and $\{\mathcal{Y}_n\}$ are *computationally indistinguishable* if for every probabilistic polynomial-time algorithm A and for sufficiently large n, $\Delta_A(\{\mathcal{X}_n\}, \{\mathcal{Y}_n\})$ is less than inverse of any positive polynomial $p(n)$.

2.3.2 One-Way Functions

2.3.1 Definition (One-way functions)

Let $\{f_n\}_{n \geq 1}$ be an ensemble of functions where $f_n : \{0,1\}^n \to \{0,1\}^{m(n)}$ is function computable in time $\mathsf{poly}(n)$. *Then we say f is one-way if for every probabilistic polynomial-time algorithm A, every positive polynomial $p(\cdot)$, and all sufficiently large n's*

$$\mathbb{P}_{x \sim \mathcal{U}_n}[A(1^n, f(x)) \in f^{-1}(f(x))] < \frac{1}{p(n)}.$$

A one-way permutation is a one-way function ensemble $\{\pi_n\}_{n \geq 1}$ where $\pi : \{0,1\}^n \to \{0,1\}^n$ is a permutation $\{0,1\}^n$.

We refer to n or 1^n as the security parameter of the one-way function. Sometimes to simplify the notation, we say $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is a one-way function to mean that the ensemble $f := \{f_n\}_{n \geq 1}, f_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is a one-way function ensemble.

We define the approximate degeneracy of a function f for a random variable X to be $D_f(X) = H(X) - H(f(X))$. We say that a one-way function f is k-regular (or k-degenerate) if in addition to the one-wayness property the number of pre-images for any $y \in \mathsf{Im}(f)$ is exactly 2^k.

We may also assume without loss of generality that the one-way functions are length-preserving, *i.e.* for all $n \in \mathbb{N}$, $f_n : \{0,1\}^n \to \{0,1\}^n$. For a reference, see [11, Chapter 2, Section 2.3.1].

2.3.3 Hardcore predicates

2.3.2 Definition (Hardcore predicate)
A polynomial-time computable predicate $b : \{0,1\}^n \to \{0,1\}$ is called a hard-core predicate of a function f if for every probabilistic polynomial-time algorithm A, every positive polynomial $p(.)$, and all sufficiently large n's

$$\mathbb{P}_{x\in\{0,1\}^n}[A(f(x)) = b(x)] < \frac{1}{2} + \frac{1}{p(n)}.$$

where the probability is taken uniformly over the possible choices of $x \in \{0,1\}^n$ and over the internal coin tosses of algorithm A .

We will use the generic hardcore predicate given by Goldreich and Levin [7].

2.3.3 Theorem. (Generic hardcore predicate : Goldreich-Levin Theorem)
Let f be any one-way function defined on input $\{0,1\}^n$ and $x, r \in \{0,1\}^n$. Then $b_r(x) = x \odot r$ is a hard-core predicate of $f(x,r) = (f(x), r)$.

Note that by taking $\log n$ many independent strings $r \in \{0,1\}^n$ in the above we can get $\log n$ bits which are computationally indistinguishable from uniform random bits even given $f(x)$ and r's.

2.3.4 Pseudorandom Generators

2.3.4 Definition (Pseudorandom Generator)
Let $\{g_n\}_{n\geq 1}$ where $g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ be a polynomial-time computable function ensemble where $\ell(n) > n$. We say that g is a Pseudorandom Generator if $g(\mathcal{U}_n)$ is computationally-indistinguishable from $\mathcal{U}_{\ell(n)}$.

Note that it is enough to consider PRGs which stretch their inputs by one bit since we can apply it repeatedly and get a PRG with arbitrarily large polynomial stretch.

2.3.5 Lemma.
Let $\{g_n\}_{n\geq 1}$ be a PRG where $g_n : \{0,1\}^n \to \{0,1\}^{n+1}$. Then for any arbitrary polynomial $p(n)$, there exists a PRG ensemble $\{g'_n\}_{n\geq 1}$ where $g_n : \{0,1\}^n \to \{0,1\}^{p(n)}$.

Using the Goldreich-Levin theorem, it is quite easy to construct pseudorandom generators from one-way permutations.

2.3.6 Theorem. (PRG from one-way permutations)
Let f be a one-way permutation. Then $g(x,r) = (f(x), r, x \odot r)$ is pseudorandom generator.

2.3.5 Black-box Constructions

A construction of a primitive P from another primitive Q consists of showing that if there exists an implementation f of Q, then there exists an implementation M_f of P. This is equivalent to showing that for every adversary that breaks M_f, there exists an adversary that breaks f.

Such a construction is called *fully-black-box* (or black-box) if there exist efficient probabilistic oracle machines M, S such that for every oracle f implementing Q we have the following :

- **Correctness:** M^f is an implementation for P.
- **Security:** For every function (adversary) G breaking the security of M^f, $S^{G,f}$ breaks the security of f.

A construction which is not fully-black-box is termed as a non-black-box construction in this thesis. However, we remark that non-black-box constructions can be classified further in terms of their limitations (c.f. [22]).

2.4 Tools from Probability

2.4.1 Chernoff Bounds

We will denote the binomial distribution with parameters n and p as $\mathsf{Bin}(n,p)$. The following is a well known tail bound for the binomial distribution which we will sometimes need in our proofs.

2.4.1 Lemma. (Chernoff Bounds)
Let X be a random variable distributed as $\mathsf{Bin}(n,p)$ and set $\mu = \mathbb{E}[X] = np$. Then, the following inequalities holds for the tails of X:

$$\mathbb{P}[X \notin [(1-\varepsilon)\mu, (1+\varepsilon)\mu]] \leq 2e^{-\frac{\varepsilon^2\mu}{3}} \qquad \text{for every } 0 < \varepsilon \leq 1.$$

$$\mathbb{P}[X \geq (1+\varepsilon)\mu] \leq \left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\mu} \qquad \text{for every } \varepsilon > 0.$$

A simple application of the above bounds gives us the following useful fact that random functions are approximately regular with high probability.

2.4.2 Lemma.
Let $f : \{0,1\}^n \to S$ be a uniform random function where $S \subseteq \{0,1\}^n, |S| = 2^{n-k}$ for an integer $k = \omega(\log n)$. Then with probability at least $1 - e^{-n^2}$, the number of pre-images for each $y \in S$ is between 2^{k-1} and 2^{k+1}.

▶ **Proof.** Let Z_y be the random variable counting the number of pre-images of y under f. Note that for any $x, y \in \{0,1\}^n$ we have $\mathbb{P}_f[f(x) = y] = 2^{-|S|} = 2^{-(n-k)}$. It follows that the random variable Z_y is distributed as $\mathsf{Bin}(2^n, 2^{-(n-k)})$. Applying the Chernoff bounds we get that

$$\mathbb{P}_f\Big[Z_y \notin [2^{-k-1}, 2^{k+1}]\Big] \leq \mathbb{P}\Big[\Big|\, Z_y - \mathbb{E}[Z_y] \,\Big| \geq \frac{1}{2}\mathbb{E}[Z_y]\Big] \leq e^{-\frac{C}{4}2^k} \leq e^{-n^3}.$$

By union bound, the probability that there is some $y \in S$ which has either less than 2^{k-1} or more than 2^{k+1} preimages is at most e^{-n^2}. ■

2.4.2 Borel-Cantelli Lemma

The following lemma will help us argue that if probability that a certain event $\mathbf{E}_n$ occurs for a bitstring of length n is small, then with probability 1, the event will happen only for finitely many lengths n of bitstrings.

2.4.3 Lemma. (Borel-Cantelli Lemma)
Let $\{\mathbf{E}_n\}_{n\geq 1}$ be a sequence of events such that $\sum_{n=1}^{\infty} \mathbb{P}[\mathbf{E}_n] < \infty$. Then it holds that

$$\mathbb{P}[\wedge_{i=1}^{\infty} \vee_{j=i}^{\infty} \mathbf{E}_j] = 0.$$

In other words, the probability that infinitely many of the events $\mathbf{E}_n$ occur is 0.

CHAPTER 3

Constructions of PRGs from One-way Functions

Before start working on lower bounds, it is a good idea to see what constructions give the best upper bounds. In line with that, here we will review the black-box non-adaptive construction of PRGs from one-way function recently discovered by Haitner, Reingold and Vadhan [14]. In fact, it was their result which inspired us to consider non-adaptive PRGs of the form $g(x_1, \ldots, x_t, f(x_1), \ldots, f(x_t))$ (where $\{f_n\}_{n\geq 1}, f : \{0,1\}^n \to \{0,1\}^n$ is the one-way function and $t \geq 1$) for lower bounds on seed lengths.

Behind the standard methods of proving the security of such constructions lies the notion of a computational analogue of entropy, usually termed as *pseudoentropy*. The notion was introduced by Håstad, Impagliazzo, Levin and Luby [10], in their breakthrough paper which first proved that the construction of PRGs from arbitrary one-way functions is possible and has turned out to be very useful in the cryptographic literature. Informally, one says that X has pseudoentropy at least k, if there exists a random variable Y which is computationally indistinguishable from X but has entropy at least k.

The fundamental step in giving a PRG construction from an arbitrary one-way function is to give a pseudoentropy generator, *i.e.* a function whose output has sufficiently more pseudoentropy than the entropy of its input. We can then take a direct product of the above pseudoentropy generator by evaluating it on a large number of independent inputs and use Lemma 2.2.4 to convert the entropy to min-entropy with a small loss per copy. If the number of copies is sufficiently large then this loss is compensated by the entropy gained by the pseudoentropy generator. The above steps essentially show that there exists a random variable which is indistinguishable from the output of the construction but which still has sufficiently large min-entropy. We can then apply the Left over hash lemma to extract the entropy and conclude that the output is pseudorandom.

Haitner, Reingold and Vadhan define a better measure of pseudoentropy, which allows them to give a much simpler and efficient constructions of PRGs from one-way functions. For a one-way function with security parameter n, their PRG achieves a seed length of $\mathcal{O}(n^4 \log n)$ and furthermore is also non-adaptive.

We will present here a weaker version of their construction which achieves seed length $\mathcal{O}(n^8 \log n)$. The essential construction is the same, but at an intermediate step they need to construct a suitable hash function family with small description. For the sake of simpler exposition here, we will work with a much simpler family of hash functions which has a much larger description length. This deteriorates the efficiency of the construction but still allows us to get the main ideas across.

We start in the next section by stating the notion of computational entropy used in [14]. In the subsequent sections, we will formally define the construction and prove its security. For brevity we will refer to their construction as the HRV construction henceforth.

3.1 Next-block Pseudoentropy

3.1.1 Definition (Next-block Pseudoentropy)
Let X be a random variable taking values in $\mathcal{U}^m$, where $X, \mathcal{U}$, and m may all depend on a security parameter n. For $k = k(n)$, we say that X has next-block pseudoentropy at least k if for every oracle-aided polynomial time distinguisher $D^{(\cdot)}$, there exists a set of random variables $Y := \{Y_1, \ldots, Y_m\}$, each over $\mathcal{U}$ such that:

1. $\sum_{i=1}^{m} H(Y_i \mid X_1, \ldots, X_{i-1}) \geq k$, *and*
2. $\mathbb{E}_{i \sim_U [m]} \left[\mathbb{P}[D^{O_{X,Y}}(X_1, \ldots, X_{i-1}, X_i) = 1] - \mathbb{P}[D^{O_{X,Y}}(X_1, \ldots, X_{i-1}, Y_i) = 1] \right] \in \mathsf{neg}(n)$ *where the oracle $O_{X,Y}(i)$, for $i \in [m]$ samples according to the joint distribution (X, Y_i).*

We say that every block of X has pseudoentropy at least $\alpha = \alpha(n)$, if condition (1) above is replaced with $H(Y_i \mid X_1, \ldots, X_{i-1}) \geq \alpha$ for every $i \in [m]$. Similarly we say that every block of X has next-block pseudo-min-entropy at least $\alpha = \alpha(n)$, if condition (1) above is replaced with $H_\infty(Y_i \mid X1, \ldots, X_{i-1}) \geq \alpha$ for every $i \in [m]$.

In the following sections, we will implicitly consider a distribution over $\{0,1\}^t$ to be a t-block distribution. Sometimes, we will also need to denote a m-block random variable where each block is distributed over $\{0,1\}^t$. We will then say that the random variable is distributed over $(\{0,1\}^t)^m$.

We say that a function $g : \{0,1\}^{s(n)} \to \{0,1\}^{m(n)}$ is a next-block pseudoentropy generator with (next-block) pseudoentropy at least k, if the random variable $g(\mathcal{U}_{s(n)})$ has next-block pseudoentropy at least k. If we talk about the next-block pseudoentropy properties of the function g, then we are referring to the random variable $g(\mathcal{U}_{s(n)})$.

3.2 The HRV Construction

The HRV construction is formally stated in Construction 3.2.1. With the right settings of parameters it yields a PRG (Theorem 3.2.2). Before proceeding with the proof of security, let us first give some intuition as to why the above construction gives a PRG. Recall that for $X \sim \mathcal{U}_n$, $D_f(X) = n - H(f(X))$ denotes the approximate degeneracy of the function. For a random variable $Q \sim \mathcal{U}_{nd}$, consider the first step of the above construction where we think of the random matrix Q as a universal hash function. The Left over hash lemma gives us that the output of g_{nb} extracts $D_f(X) - c \log n$ bits of entropy from X given $f(X)$ and Q for some constant $c > 0$. Then the Goldreich-Levin theorem

(Theorem 2.3.3) gives us additional $(c+1)\log n$ hard-core bits. In particular, one can use this to show that the output of g_{nb} has next-block pseudoentropy at least $k := H(f(\mathcal{U}_n), Q) + \log n$.

Note that the original proof in [14] uses a different hash function family than the family of boolean matrices as used above. This family of hash functions is not 2-universal but slightly weaker, but it has description length only $\mathcal{O}(n)$ instead of $\mathcal{O}(n^2)$ as above. However, they are able to prove that it still yields a next-block pseudoentropy generator with pseudoentropy at least k. This allows one to achieve much better parameters and get a seed length of $\mathcal{O}(n^4 \log n)$. However as mentioned before, here we will try to keep things simple and work with the hash function family of boolean matrices.

Proceeding further, the entropy equalization step then spreads out this pseudoentropy uniformly and gives us a random variable where each bit has at least k/m bits of next-block pseudoentropy relative to X where $m = n + nd + d$ is the output length of g_{nb}. The direct product then flattens the next-block pseudoentropy to next-block pseudo-min-entropy. By talking sufficiently many copies we can ensure that the loss due to flattening is compensated by the additional entropy gained due to the next-block pseudoentropy generator. Finally, an application of Left over hash lemma then implies that the output is pseudorandom.

▶ **Construction. 3.2.1**
Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function. Let $\delta = \delta(n), d = d(n), \ell = \ell(n), t = t(n), \Gamma = \Gamma(n)$ and $\kappa = \kappa(n)$ be parameters. Define the following functions.

- ***(Next-block Pseudoentropy Generator)*** *Let $g^f_{nb}(x,q) = (f(x), q, q(x))$ where $q(x) = x \odot q$ by interpreting $q \in \{0,1\}^{n\times d}$ as a $n \times d$ boolean matrix and $x \in \{0,1\}^n$ as a $1 \times n$ row vector.*
- ***(Entropy Equalization)*** *Set $m = n + nd + d$. For $j \in [m]$ and $u_1, \ldots, u_\ell \in \{0,1\}^m$, define $eq(j, u_1, \ldots, u_\ell) = (u_1|_{\{j,\ldots,m\}}, u_2, \ldots, u_{\ell-1}, u_\ell|_{\{j-1,\ldots,m\}})$.*
- ***(Direct Product and Entropy Extraction)*** *Set $\alpha = n + nd + \delta$ and $m' = m(\ell - 1)$. Define $(z_1, \ldots, z_t) = eq^{\otimes t}(j, g^f_{nb}(x,q))$ where $z_i = eq(j_i, g^f_{nb}(x_i, q_i))$ for $j \in (\{0,1\}^{\log n})^t, x \in (\{0,1\}^n)^t$ and $q \in (\{0,1\}^{n\times d})^t$. For $i \in [m']$, let w_i denote the string $(z_1|_{\{i\}}, z_2|_{\{i\}}, \ldots, z_t|_{\{i\}})$. Finally define the function $g : \{0,1\}^{(n+nd)\ell t} \to \{0,1\}^{(t+t\alpha-\Gamma-\kappa)m'}$:*

$$g^f(x,q,j,s) = (s(w_1), s(w_2), \ldots, s(w_{m'})),$$

where s is interpreted as a universal hash function from $\{0,1\}^t$ to $\{0,1\}^{t\alpha-\Gamma-\kappa}$.

▶ **Theorem. 3.2.2**
Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function. There exists constants $c_1, c_2 > 0$ such that for $\delta = \log n, d = n + 5\log n, \ell = \frac{c_1(n+nd)}{\log n}, t = (n+nd+d)^2 \log^2 n, \Gamma = c_2\sqrt{t \log n} \cdot \log t$ and

$\kappa = \log^2 n$, g^f as defined in Construction 3.2.1, is a pseudorandom generator with seed length $(n + nd)\ell t = \mathcal{O}(n^8 \log n)$.

3.3 Proof of Security of HRV Construction

Let us start with the proof of the security of the HRV construction. To keep the notation simpler we will prove the statements for a slightly more general settings of parameters.

First we will prove that g_{nb} as defined in Construction 3.2.1 is a next-block pseudoentropy generator with pseudoentropy at least $n + nd + \log n$. The basis for proving this is the following lemma.

3.3.1 Lemma.
Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function and $g_{nb} : \{0,1\}^n \to \{0,1\}^m$ be as defined in Theorem 3.2.2 with $m = n + nd + d$. Let I be an integer uniformly drawn from $\{0, \ldots, d-1\}$. Define the functions $g'(x,q) = (f(x), q, q(x)|_{\{1,\ldots,I\}})$ and the predicate function $p : \{0,1\}^n \to \{0,1\}$, $p(x,q) = q(x)|_{\{I+1\}}$. Then the following holds for all PPT algorithms P and $X \sim \mathcal{U}_n$ and $Q \sim_{\mathcal{U}} \{0,1\}^{n \times d}$

$$\mathbb{P}_{P,I}[P(g'(X,Q)) = p(X,Q)] \leq \frac{1}{2} + \frac{D_f(X) + \log n}{2d}.$$

▶ **Proof.** Note that the Left over hash lemma (Lemma 2.2.3) implies that given $y = f(X)$ and Q, the first $K := D_f(X) - 4\log n$ bits of $Q(X)$ have statistical distance at most $n^{-3/2}$ from uniform random bits. Also using the Goldreich-Levin theorem (Theorem 2.3.3) we get that for any PPT algorithm the advantage in distinguishing $(f(X), Q, Q(X))$ from $(f(X), Q, Q(X)|_{\{1,\ldots,K\}}, \mathcal{U}_b, Q(X)|_{\{K+l+1,\ldots,d\}})$ where $l = 6\log n$, is negligible. It follows that no PPT algorithm can distinguish distinguish $(f(X), Q, Q(X))$ from $(f(X), Q, R_1, \ldots, R_{K'}, Q(X)|_{\{K'+1,\ldots,d\}})$ with advantage more than $1/n$ where $K' = K + l$ and $R_1, \ldots, R_{K'}$ are uniform and independent random bits. Therefore, we have for any PPT algorithm P

$$\begin{aligned}
&\sum_{i=0}^{d-1} \mathbb{P}[P(f(X), Q, Q(X)|_{\{1,\ldots,i-1\}}) \neq Q(X)|_{\{i+1\}}] \\
&\geq \sum_{i=0}^{d-1} \mathbb{P}[i < K' \wedge (P(f(X), Q, Q(X)|_{\{1,\ldots,i-1\}}) \neq Q(X)|_{\{i+1\}})] \\
&\geq \sum_{i=0}^{d-1} (\mathbb{P}[i < K' \wedge (P(f(X), Q, R_1, \ldots, R_{i-1}) \neq R_{i+1})] - d/n) \\
&\geq \sum_{i=0}^{d-1} (\mathbb{P}[i < K']/2 - d/n) \\
&= \mathbb{E}[K']/2 - d/n \\
&\geq \frac{D_f(X) + \log n}{2},
\end{aligned}$$

which finishes the proof of the lemma. ■

With the proof of the above lemma in hand, we are done with respect to non-uniform security because for the random variable $Y := (f(X), Q, Q(X)|_{\{1,\ldots,K\}}, \mathcal{U}_b, Q(X)|_{\{K'+1,\ldots,d\}})$ constructed in the proof above, we have

$$\begin{aligned}\sum_{i=1}^{m} H(Y_1 \mid X_1 \ldots X_m) &= H(f(S)) + H(Q) + H(R_1, \ldots, R_{K'} \mid f(S), Q) \\ &= H(f(S)) + H(Q) + D_f(X) + \log n = n + nd + \log n\end{aligned}$$

where m, K and K' are as defined in the proof.

But relative to uniform security we have a problem since the random variable Y might not be efficiently samplable as $D_f(X)$ might not be efficiently computable[1]. To elaborate more, to achieve security with respect to uniform adversaries, we need to show that given an efficient algorithm to break the final PRG we can use it to construct another efficient algorithm to break the next-block pseudoentropy of g_{nb}. But the efficiency of sampling the random variable Y is a problem. However, employing the uniform hard-core lemma [16, 2] allows us to define efficiently samplable random variables from the above while still retaining at least $H(f(S), Q) + \log n$ bits of next-block pseudoentropy.

3.3.2 Lemma. (Uniform Hard Core Lemma - Entropy Version [14])
Let n be a security parameter, $\delta_0 = \delta_0(n) \in [0,1]$, $\delta = \delta(n) \in [\delta_0, 1]$, and $\gamma = \gamma(n) \in [0,1] > 2^{-n/3}$. Let (A,B) be a polynomial time samplable random variable over $\{0,1\}^n \times \{0,1\}$ such that for every probabilistic algorithm M running in time $T = T(n)$ and large enough n

$$\mathbb{P}[M(\delta_0, \gamma, A) = B] \leq 1 - \delta/2.$$

Then for every oracle-aided distinguisher $D^{(\cdot)}$ running in time $T = T \cdot (\delta_0\gamma/n)^{\mathcal{O}(1)}$ and all sufficiently large n, there is a random variable C, jointly distributed with (A,B), such that:

1. $H(C \mid A) \geq \delta$.
2. $\mathbb{P}[D^{O_{A,B,C}}(A,B) = 1] - \mathbb{P}[D^{O_{A,B,C}}(A,C) = 1] \leq \gamma$, *where $O_{A,B,C}$ is an oracle that samples according to the joint distribution (A,B,C).*

We now use the above lemma to prove that g_{nb} is a next-block pseudoentropy generator even for uniform adversaries.

3.3.3 Lemma.
Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function. Then g_{nb}^f as defined in Theorem 3.2.2 is a next-block pseudoentropy generator with pseudoentropy at least $n + nd + \log n$. Moreover, the reduction from security of g_{nb} to that of f is fully black-box.

[1] In the case of non-uniform security we can hardwire the samples.

▶ **Proof. (Adapted from [14])** Let $W = (S, Q)$ be uniformly distributed over $\{0,1\}^n \times \{0,1\}^{n \times d}$, let $X = g_{nb}(W)$ and $m = n + nd + d$. Assume for the sake of contradiction, that there exists a PPT distinguisher D_{nb} which can next-block distinguish every random variable Y over $\{0,1\}^m$ satisfying $\sum_{i=1}^{m} H(Y_i \mid X_1, \ldots, X_{i-1}) \geq n + nd + \log n$, with an advantage δ which is infinitely often greater than $\varepsilon(n)$ where $\varepsilon(n)$ is inverse polynomial in n.

Note that $X = (f(S), Q, Q(S)|_{\{1,\ldots,d\}})$. For $I \sim_{\mathcal{U}} \{0, \ldots, d-1\}$, define $A = (f(S), Q, Q(S)|_{\{1,\ldots,I\}})$ and $B = Q(S)|_{\{I+1\}}$. Then Lemma 3.3.1 implies that for any PPT algorithm M the following holds:

$$\mathbb{P}[M(A) = B] \leq 1 - \frac{\beta}{2} \tag{3.1}$$

for $\beta = \frac{D_f(S) + \log n}{n}$.

Let C be any random variable, jointly distributed with A such that $H(C \mid A) \geq \delta$. Using this we define a random variable Y over $\{0,1\}^m$ as follows:

$$Y_i|_{X=(x_1,\ldots,x_m)} = \begin{cases} x_i & \text{if } i \leq d + nd \\ C(S,H)|_{I=i,\ X=(x_1,\ldots,x_m)} & \text{otherwise.} \end{cases}$$

Then, the next-block entropy of Y relative to X is at least $n + nd + \log n$:

$$\begin{aligned} \sum_{i=1}^{m} H(Y_1 \mid X_1 \ldots X_m) &= \sum_{l=1}^{n'} H(X_i \mid X_1 \ldots X_{i-1}) + \sum_{i=1}^{n} H(C \mid A, I = i) \\ &= H(f(S), Q) + n \cdot H(C \mid A) \\ &\geq H(f(S)) + H(Q) + (n - H(f(S)) + \log n) \\ &= n + nd + \log n. \end{aligned}$$

Note that this means that we can use the distinguisher D_{nb} to next-block distinguish the above Y from X with advantage δ. Also note that D_{nb} must gain its entire advantage when $i > n + nd$ since otherwise $Y_i = X_i$. Reinterpreting it, this means that the distinguisher D_{nb} is distinguishing (A, B) from (A, C). Moreover, the oracle queries made by D_{nb} to $O_{X,Y}$ can be simulated with queries to the oracle $O_{A,B,C}$. (It might have to make $\mathcal{O}(n)$ queries to get a sample with the desired value of i with probability at least $1/2$.) This means that we get a polynomial time algorithm which distinguishes (A, B) from (A, C) with distinguishing advantage $\delta/2$ which is infinitely often greater than $\varepsilon(n)/2$ which we recall is inverse polynomial in n.

Proposition 3.3.2 then implies that there exists a PPT algorithm which can predict B from A with success probability strictly larger than $1 - \beta/2$ for infinitely many n. (We can fix $\delta_0 = 1/n$ and $\gamma = \varepsilon(n)/2$ in Proposition 3.3.2 since $\beta \geq 1/n$ always.) ■

The entropy equalization step gives us a better handle on how the next-block pseudoentropy is distributed across various blocks.

3.3.4 Lemma. (Entropy Equalization)
Let n be a security parameter and $m = m(n) \in \mathsf{poly}(n), \ell = \ell(n) \in \mathsf{poly}(n)$ be polynomial time computable functions. Let X be a random variable over $\{0,1\}^m$ with next-block pseudoentropy at least $k = k(n)$. Let $J \sim_{\mathcal{U}} [m]$ and let $\tilde{X} = eq(J, X^{(1)}, \ldots, X^{(\ell)})$ where $X^{(i)}$ are i.i.d. copies of X and $eq(\cdot)$ is the function defined in Construction 3.2.1. Then every block of $\tilde{X}$ has next-block pseudoentropy at least k/m. Moreover, the reduction between security of $\tilde{X}$ and X is fully black-box.

▶ **Proof. (Adapted from [14])** Set $m' = m(\ell - 1)$ and let $Y = \{Y_1, \ldots, Y_m\}$ be a random variable jointly distributed with X. Similarly let $Y^{(1)}, \ldots, Y^{(\ell)}$ be i.i.d. copies of Y. Let $\tilde{Y} = eq(J, Y^{(1)}, \ldots, Y^{(\ell)})$ be jointly distributed with $\tilde{X}$ where J is the value used in computing $\tilde{X}$ and $X^{(j)}$ is jointly distributed with $Y^{(j)}$ as per the distribution (X, Y). Note that $\tilde{Y}_i \sim Y_{J+i-1 \mod m}$ (where we define $m \mod m = m$ rather than 0) and $J + i - 1 \sim_{\mathcal{U}} [m]$.

Then, for every $i \in [m']$ we have

$$\begin{aligned} H(\tilde{Y}_i \mid \tilde{X}_1 \ldots \tilde{X}_{i-1}) &\geq H(Y_{J+i-1 \mod m} \mid X_1 \ldots X_{J+i-1 \mod m}) \\ &= \mathbb{E}_{i' \sim_{\mathcal{U}} [m]}[H(Y_{i'} \mid X_1 \ldots X_{i'-1}] \\ &\geq k/m. \end{aligned} \tag{3.2}$$

Now we show that any PPT next-block distinguisher $\tilde{D}$ for $\tilde{X}$ with non-negligible distinguishing advantage can be used to give a PPT distinguisher D for next-block pseudoentropy of X. The distinguisher D works as follows: on input $(x_1, \ldots, x_{i-1}, z)$, it samples $x' = (x'_1, \ldots, x'_{m'})$ distributed according to $\tilde{X}$ using its oracle access to $O_{X,Y}$. Let j be the value of J used to generate x'. It then selects $i' \in [m']$ uniformly at random conditioned on $i' = j + i - 1 \mod m$ and then returns the value $\tilde{D}^{O_{\tilde{X},\tilde{Y}}}(x'_1, \ldots, x'_{i-1}, x_1, \ldots, x_{i-1}, z)$ while using its oracle access to $O_{X,Y}$ to simulate queries to $O_{\tilde{X},\tilde{Y}}$ (which requires only a polynomial overhead).

It is easily seen that if $\tilde{D}$ runs in polynomial time, then so does D. Also for any variable Y distributed jointly with X as above with $\sum_{i \in [m]} H(Y_i \mid X_1, \ldots, X_{i-1}) \geq k$, Equation (3.2) yields that the distribution of the query to $\tilde{D}$ made by D is identical to the one used for distinguishing the next-block pseudoentropy of $\tilde{X}$ from $\tilde{Y}$. It follows that the distinguishing advantage of D is the same as that of $\tilde{D}$ on a random challenge and hence is non-negligible. ■

Now we show how one can transform the next-block pseudoentropy to next-block pseudo-min-entropy by taking a direct product albeit for a small loss.

3.3.5 Lemma. (Direct Product)
Let n be a security parameter and $m = m(n) \in \mathsf{poly}(n), t = t(n) \in \mathsf{poly}(n), t(n) \geq n$ be polynomial time computable parameters. Let X be a random variable over $\{0,1\}^m$ where every block of X has next-block pseudoentropy at least $\alpha = \alpha(n)$. Define the t-wise direct product of X as

$$X^{\otimes t} = ((X_1^{(1)}, \ldots, X_1^{(t)}), \ldots, (X_m^{(1)}, \ldots, X_m^{(t)})).$$

where $X^{(1)}, \ldots, X^{(t)}$ are i.i.d. copies of X. We consider $(X_i^{(1)}, \ldots, X_i^{(t)})$ for $i \in [m]$ as the blocks of $X^{\otimes t}$.

Then for every $\kappa = \kappa(n) \geq \log^2 n$, every block of $X^{\otimes t}$ has next-block pseudo-min-entropy at least $\alpha' = t\alpha - \mathcal{O}(\sqrt{t\kappa} \log t)$.

▶ **Proof. (Adapted from [14])** For any random variable $Y = \{Y_1, \ldots, Y_m\}$ distributed jointly with X over $\{0,1\}^m$, define the variable $Y^{\otimes t} = Y^{(1)}, \ldots, Y^{(t)}$ jointly distributed with $X^{\otimes t}$ where $Y^{(j)}, j \in [t]$ are i.i.d. copies of Y. Furthermore, for $i \in [m]$, we will use the notation $Y_i^{\otimes t}$ to denote $(Y_i^{(1)}, \ldots, Y_i^{(t)})$ or the i^{th} block of $Y^{\otimes t}$.

Given an adversary D_t which breaks the next-block pseudo-min-entropy of of $X^{\otimes t}$, we define the following distinguisher$D$ to break the next-block pseudoentropy of X as follows : given an input $(x_1, \ldots, x_{i-1}, z)$ and oracle access to $O_{X,Y}$, D first samples $j \sim_{\mathcal{U}} [t]$ and $(x^{\otimes t}, y^{\otimes t}) \sim (X^{\otimes t}, Y^{\otimes^t})$. Then it replaces $(x_1^{(j)}, \ldots, x_{i-1}^{(j)})$ with $(x_1, \ldots, x_{i-1})$, sets $z^{[j]} = (x_i^{(1)}, \ldots, x_i^{(j-1)}, z, y_i^{(j+1)}, \ldots, y_i^{(t)})$ and returns $D_t^{O_{X^{\otimes t}, Y^{\otimes t}}}(x_1^{\otimes t}, \ldots, x_{i-1}^{\otimes t}, z^{[j]})$ while simulating queries to $O_{X^{\otimes t}, Y^{\otimes t}}$ with $O_{X,Y}$.

It is easily seen that if D_t runs in polynomial time, then so does D as the overhead to simulate oracle queries is polynomial.

If $H(Y_i \mid X_1, \ldots, X_{i-1}) \geq \alpha$ for $i \in [m]$, then Lemma 2.2.4 implies that there exists a random variable $W = W_1, \ldots, W_m$ over $(\{0,1\}^t)^m$ jointly distributed with $X^{\otimes t}$ such that the following hold:

- $\|(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, Y_i^{\otimes t}) - (X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, W_i)\| \leq 2^{-\kappa} + 2^{-\Omega(t)} \in \mathsf{neg}(n)$.
- $H_\infty(W_i \mid x_1^{\otimes t}, \ldots, x_{i-1}^{\otimes t}) \geq t\alpha - \mathcal{O}(\sqrt{t\kappa}\log t)$ for every $x \in \mathsf{Supp}(X^{\otimes t})$.

For $j \in [t]$, define the hybrid distribution $Z^{[j]} = (X_i^{(1)}, \ldots, X_i^{(j)}, Y_i^{(j+1)}, \ldots, Y_i^{(t)})$. Then for each $i \in [m']$, we have

$$
\begin{aligned}
&\delta^D_{X,Y,i} \\
&:= \frac{1}{t}\sum_{j\in[t]}(\mathbb{P}[D_t^{O_{X^{\otimes t},Y^{\otimes t}}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, Z^{[j]}) = 1] - \mathbb{P}[D_t^{O_{X^{\otimes t},Y^{\otimes t}}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, Z^{[j-1]}) = 1]) \\
&= \frac{1}{t}(\mathbb{P}[D_t^{O_{X^{\otimes t},Y^{\otimes t}}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, Z^{[t]}) = 1] - \mathbb{P}[D_t^{O_{X^{\otimes t},Y^{\otimes t}}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, Z^{[0]}) = 1]) \\
&= \frac{1}{t}(\mathbb{P}[D_t^{O_{X^{\otimes t},Y^{\otimes t}}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, X_i^{\otimes t}) = 1] - \mathbb{P}[D_t^{O_{X^{\otimes t},Y^{\otimes t}}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, Y_i^{\otimes t}) = 1]) \\
&\geq \frac{1}{t}(\mathbb{P}[D_t^{O_{X^{\otimes t},Y^{\otimes t}}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, X_i^{\otimes t}) = 1] - \mathbb{P}[D_t^{O_{X^{\otimes t},W_i}}(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, W_i) = 1]) - \mathsf{poly}(n)(2^{-\kappa} + 2^{-\Omega(t)}) \\
&:= \frac{1}{t}(\delta^{D_t}_{X^{\otimes t},W_i} - \mathsf{neg}(n)),
\end{aligned}
$$

where in the second last line the factor of $\mathsf{poly}(n)$ comes from the fact that D_t can make at most polynomially many oracle queries to distinguish $(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, Y_i^{\otimes t})$ and $(X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}, W_i)$.

Taking an expectation over $i \in [m]$, we get that the advantage $\delta^D_{X,Y}$ that D has in next-block distinguishing X and Y satisfies,

$$\delta^D_{X,Y} \geq \frac{1}{t}(\delta^{D_t}_{X^t,W} - \mathsf{neg}(n)),$$

where $\delta^{D_t}_{X^t,W}$ is the advantage D_t has in next-block distinguishing $X^{\otimes t}$ and $W = \{W_1, \ldots, W_m\}$. Note that for any Y distributed jointly with X over $\{0,1\}^m$ satisfying $H(Y_i \,|X_1, \ldots, X_{i-1}) \geq \alpha$ for every $i \in [m]$, by definition W as above satisfies $H(W_i \,|X_1^{\otimes t}, \ldots, X_{i-1}^{\otimes t}) \geq \alpha$ for every $i \in [m]$ and hence $\delta^{D_t}_{X^t,W}$ is non-negligible since D_t distinguishes the next-block pseudoentropy of W from $X^{\otimes t}$. It then follows that $\delta^D_{X,Y}$ is also non-negligible which is a contradiction to the next-block pseudoentropy of X. ■

In the final step, we use a universal hash function to extract the next-block pseudo-min-entropy to get a pseudorandom string.

3.3.6 Lemma. (Entropy Extraction)
Let n be a security parameter and $m = m(n) \in \mathsf{poly}(n)$, $t = t(n) \in \mathsf{poly}(n)$, $\alpha = \alpha(n) \in [t(n)]$ and $\kappa = \kappa(n)$ with $\log^2 n \leq \kappa \leq \alpha$. Let X be a random variable over $(\{0,1\}^t)^m$ such that every block of X has next-block pseudo-min-entropy at least α. For $x \in (\{0,1\}^t)^m, s \in \{0,1\}^t$ define the function

$$Ext(x,s) = (s, s(x_1), \ldots, s(x_m)),$$

where s is interpreted as a universal hash function from $\{0,1\}^t$ to $\{0,1\}^{m(\alpha-\kappa)}$.[2]

Then $Ext(X, \mathcal{U}_t)$ is pseudorandom with a black-box proof of security.

▶ **Proof. (Adapted from [14])** Given any polynomial time distinguisher D which breaks the pseudorandomness of $Ext(X, \mathcal{U}_t)$, we can easily build a polynomial time distinguisher D_{PRG} which breaks the next-block pseudo-min-entropy of X as follows : on input $(x_1, \ldots, x_{i-1}, z)$ it computes $(s, s(x_1), \ldots, s(x_{i-1}), s(z), U_{(\alpha-\kappa)(m-i)})$ and queries D on it and returns the answer to this query as its output.

Define the hybrid distribution $Z^{[i]}(W) = (S, S(X_1), \ldots, S(X_{i-1}), S(W), U_{(\alpha-\kappa)(m-i)})$ where Q is a uniformly distributed hash function. Note that $Z^{[m]}(X_m)$ denotes the output distribution of $Ext(X, U_t)$ while $Z^{[0]}(X_0)$ is the uniform distribution on $\{0,1\}^{t+m(\alpha-\kappa)}$. Let $Y = \{Y_1, \ldots, Y_m\}$ be any random variable, where Y_i's are defined over $\{0,1\}^t$, which is jointly distributed with X such that for every $i \in [m]$ and every $x \in \mathsf{Supp}(X)$, $H_\infty(Y_i \mid X_1, \ldots, X_{i-1} = x_1, \ldots, x_{i-1}) \geq \alpha$. Then we can write the following about the distinguishing advantage δ_D for the distinguisher D :

$$\begin{aligned}
\delta_D &= \mathbb{P}[D_{PRG}(Z^{[m]}(X_m)) = 1] - \mathbb{P}[D_{PRG}(Z^{[0]}(X_0)) = 1] \\
&= \sum_{i=1}^{m} \mathbb{P}[D_{PRG}(Z^{[i]}(X_i)) = 1] - \mathbb{P}[D_{PRG}(Z^{[i-1]}(X_{i-1})) = 1] \\
&= \sum_{i=1}^{m} (\mathbb{P}[D_{PRG}(Z^{[i]}(X_i)) = 1] - \mathbb{P}[D_{PRG}(Z^{[i]}(Y_i)) = 1]) + \sum_{i=1}^{m} \|Z^{[i]}(Y_i) - Z^{[i-1]}(X_{i-1})\|
\end{aligned}$$

Note that the first term above is exactly the distinguishing advantage of next-block distinguisher D_{PRG} when the samples are drawn from the joint distribution (X, Y) while the second term is at most $2^{-\kappa/2} \in \mathsf{neg}(n)$ due to the Left over hash lemma (Lemma 2.2.3). In particular, this implies that if the distinguishing advantage of D_{PRG} is non-negligible, then D is a next-block pseudo-min-entropy distinguisher for X with non-negligible distinguishing advantage. ■

With the above lemmas in hand we can now prove Theorem 3.2.2.

▶ **Proof. (Theorem 3.2.2)** Lemma 3.3.3 gives us that g_{nb} is a next-block pseudoentropy generator with pseudoentropy at least $k := n + nd + \log n$. Set $s = d + nd$ and $m = d + nd + d$ to be the seed and output lengths of g_{nb} respectively and let $X = g_{nb}(\mathcal{U}_s)$.

[2] For example, we can take $s(x) = s \odot x$ over $GF(2)$.

Next we apply Lemma 3.3.4 with $m = n+nd+d$ and $\ell = c_1 s/m$ to obtain $\tilde{X} = eq(J, X^{(1)}, \ldots, X^{(\ell)})$ where J is uniformly distributed in m and $X^{(i)}$'s are i.i.d. copies of X. This gives that every block of $\tilde{X}$ has next-block pseudoentropy at least $\alpha := k/m$.

Taking the Direct product of $\tilde{X}$ we obtain $\tilde{X}^{\otimes t}$ for $t = m^2 \log^2 n$. Lemma 3.3.5 with $\kappa = \log^2 n$ then implies that each block of $\tilde{X}^{\otimes t}$ has next-block pseudo-min-entropy at least $t\alpha - \Gamma$ where $\Gamma = \mathcal{O}(\sqrt{t \log n} \log t)$.

Finally the extraction step (Lemma 3.3.6) with $\kappa = \log^2 n$ yields that the output of g is pseudorandom. The only thing left to check is the fact that the output length is strictly larger than the seed length.

The seed length for the construction is easily seen to be $s\ell t$, while the output length for some suitable constants c_1, c_2 and c_3 is given by

$$\begin{aligned} r &:= (t\alpha - \Gamma - \kappa)m(\ell - 1) \\ &= s\ell t + t\ell \log n - st - m\Gamma\ell + \mathcal{O}(n^4 \log^2 n) \\ &\geq s\ell t + c_1 st - st - c_2 m\ell\sqrt{t}\log^2 n + c_3 n^4 \log^2 n \\ &= s\ell t + (c_1 - c_2 - 1)n^6 \log^3 n + c_3 n^4 \log^2 n \\ &= s\ell t + \Omega(n^6). \end{aligned}$$

■

CHAPTER 4

Techniques for Black-box Separation

Black-box constructions of cryptographic primitives from one-way functions (or any other primitive) relativize, *i.e.*, the proof of security holds even if one considers adversaries with access to an arbitrarily powerful oracle. To see this one has to look closely at the definition of black-box constructions : let us suppose we have a fully black-box construction for a cryptographic primitive g^f from a one-way function f. With reference to security this means that any adversary G which breaks the security of g^f can be used to construct an adversary $F^{G,f}$ (with black-box or oracle access to G and f) such that F inverts the one-way function. F might do arbitrary computations in its limited time, might query f or G and use the answers of these queries to return a possible pre-image. This implies that the only way the reduction depends on G is with respect to the guarantee that G breaks the security of g. If both G and F have access to an oracle O then the constructed adversary $F^{G^{(\cdot)},\mathsf{O},f}$ can proceed in the same way as before using its oracle access to O only to simulate the queries of G to O. Again any answer returned by a query of F^{O} to G^{O} satisfies the same guarantee as before and all the other steps remain the same, so the proof of security still holds.

Consequently, if one wants to show that certain black-box constructions of primitives from one-way functions are not possible, then it is sufficient to show there exists an oracle (called the *separation oracle*) relative to which the security of the primitive is broken but one-way functions remain secure. In particular, if such constructions exist then they have to be non-relativizing (hence must be non black-box).

In this chapter we will illustrate the standard techniques for black-box separation of cryptographic primitives from one-way permutations. We will show how to construct a separation oracle for a very simple construction and then review the techniques which can be used to show that there exists a hard function with respect to the separation oracle. Later we shall employ these methods to prove lower bounds on the seed length for a particular class of black-box PRG constructions from one-way functions.

Before proceeding further, it is important to realize what black-box separation achieves in the end. Saying that the black-box construction of a certain cryptographic primitive, say a PRG g of a special form, from one-way functions is impossible only means that there exists one such secure function f such that if we give f as an oracle to g then the result is not secure. It might very well be that f is not efficiently computable and hence not one-way. However, it is still entirely within the realms

of possibilities that the function g in question is a PRG if one restricts only to the case of one-way functions. But in this case g must use the internal implementation details of f. Such non black-box constructions fall outside the scope of black-box separation. In general, proving impossibility results for non black-box constructions is at least as hard as proving $\mathsf{P} \neq \mathsf{NP}$. In this thesis however, we will only be concerned with black-box impossibility results.

We now proceed with an example for black-box separation.

4.1 A Simple Black-box Impossibility Result

Since the main focus of this thesis is constructions of PRGs from One-way functions, here we will try to prove that the following simple black-box construction from one-way permutation is not a PRG.

4.1.1 Theorem.
There does not exist any such oracle function ensemble $\{g_n\}_{n\geq 1}, g^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+k}$ *such that the following holds for every permutation ensemble* $\{\pi_n\}_{n\geq 1}, \pi : \{0,1\}^n \to \{0,1\}^n$ *for infinitely many* n *:*

- $g_n^{\pi_n}(x) = (\pi_n(x), x|_{\{1,\ldots,k\}})$ *where* $k = k(n) = \omega(\log n)$.
- *If* π *is secure, then* $g_n^{\pi_n}(\mathcal{U}_n)$ *is computationally indistinguishable from* $\mathcal{U}_{n+k}$ *with a black-box proof of security.*

We start by defining the separation oracle in the next section and then showing that it satisfies the properties we want in the subsequent sections.

4.1.1 The separation oracle

Consider the following oracle Breaker formally described in Algorithm 1. The oracle when given an input $(1^n, w)$ where $w \in \{0,1\}^{n+k}$ goes through all $x \in \{0,1\}^n$, applies π_n on x, and checks if the concatenation of $\pi_n(x)$ and first k bits of x equals w. Note that the oracle Breaker appears to be quite powerful but we shall later show that it is insufficient to successfully invert all one-way permutation ensembles $\{\pi_n\}_{n\geq 1}$.

Algorithm 1: $\mathsf{Breaker}^{\pi}(1^n, w)$:

```
for all x ∈ {0,1}^n do
    if (π_n(x), x|_{1,...,k}) = w then
        return x;
return ⊥
```

4.1.2 Breaking the PRG construction

Let $\{\pi_n\}_{n\geq 1}$ be an arbitrary permutation ensemble. Given the definition of $\mathsf{Breaker}^{\pi}$, it is trivial to construct an adversary which distinguishes the output of $g_n^{\pi_n}(\mathcal{U}_n)$ from $\mathcal{U}_{n+k}$ for all large enough n. The adversary G^{Breaker} receives as input $(1^n, w)$ where $w \in \{0,1\}^{n+k}$ and queries Breaker with it. If Breaker outputs $\perp$, then G outputs 0 otherwise it outputs 1.

4.1.2 Lemma.
Let $\{\pi_n\}_{n\geq 1}$ be an arbitrary permutation ensemble and let $G^{\mathsf{Breaker}}(\cdot)$ be the distinguisher defined as above. Then for large enough n,

$$\left| \mathbb{P}[G^{\mathsf{Breaker}^{\pi}}(1^n, g(\mathcal{U}_n)) = 1] - \mathbb{P}[G^{\mathsf{Breaker}^{\pi}}(1^n, \mathcal{U}_{n+k}) = 1] \right| \geq \frac{1}{2}.$$

▶ **Proof.** In case G^{Breaker} gets $(1^n, w)$ as input where $w := g(x)$ for $x \sim \mathcal{U}_n$, it is clear that it always returns 1. On the other hand, note that since the adversary G^{Breaker} is deterministic, the probability that it outputs 1 on $(1^n, w)$ where $w \sim \mathcal{U}_{n+k}$ is at most $|\mathsf{Im}(g)|/2^{n+k} \leq 2^{-k} \leq 1/2$ for large enough n since $k = \omega(\log n)$. ■

4.1.3 Existence of hard permutations relative to Breaker

We will now prove that Breaker is not too powerful, *i.e.* there exists a permutation ensemble $\{\pi_n\}_{n\geq 1}$ such that no adversary with oracle access to Breaker can successfully invert the permutations for infinitely many n. The proof proceeds in two stages. In the first stage, we show that a random permutation is secure in the presence of Breaker :

4.1.3 Lemma.
Let $\{\pi_n\}_{n\geq 1}$ be a function ensemble where $\pi_n : \{0,1\}^n \to \{0,1\}^n$ is a random permutation and let $F^{\mathsf{Breaker},\pi}$ be any probabilistic polynomial time adversary that takes as input $(1^n, y)$ where $y \in \{0,1\}^n$ and has oracle access to Breaker *and π. Then there exists a negligible function $\varepsilon'(n)$ such that the following holds.*

$$\mathbb{P}_{\pi, y\sim\mathcal{U}_n}[F^{\mathsf{Breaker},\pi}(1^n, y) = \pi_n^{-1}(y)] \leq \varepsilon'(n).$$

Without loss of generality, we may focus our attention only to deterministic polynomial time adversaries in the above lemma. To see this, let δ be the success probability of a PPT adversary F in inverting a random y. Let δ_r denote the advantage of F when the random coins in F are fixed to r. Note that we have $\mathbb{E}_r[\delta_r] \geq \delta$. This implies that there exists a fixed value r_0 for random coins such that $\delta_{r_0} \geq \delta$. By fixing the random coins used by F to r_0 we get a deterministic polynomial time adversary with success probability at least δ. Consequently, it suffices to prove Lemma 4.1.3 only for deterministic adversaries.

With the above lemma in hand, the next step is to show that there exists a secure permutation ensemble $\{\pi_n\}_{n\geq 1}$ with respect to all PPT adversaries. This follows by applying the Borel-Cantelli Lemma. The formal statement is as follows.

4.1.4 Lemma.
There exists a negligible function $\varepsilon(n)$ such that with probability 1 over the choice of permutation ensembles $\{\pi_n\}_{n\geq 1}$ where $\pi_n : \{0,1\}^n \to \{0,1\}^n$ is a random permutation, the following holds for all probabilistic polynomial time adversaries $F^{\mathsf{Breaker},\pi}$ (with oracle access to Breaker *and π) and large enough n.*

$$\mathbb{P}_{y\sim\mathcal{U}_n}[F^{\mathsf{Breaker},\pi}(1^n, y) = \pi_n^{-1}(y)] \leq \varepsilon(n).$$

The above lemma together with Lemma 4.1.2 establishes Theorem 4.1.1.

Next we show how to prove the above two lemmas.

4.1.4 Hardness of Random Permutations relative to Breaker

There are two standard methods of proving Lemma 4.1.3. First one is a technique by Gennaro and Trevisan [5] where one shows that if a polynomial-time adversary with access to the separation oracle Breaker inverts a random permutation $\pi_n : \{0,1\}^n \to \{0,1\}^n$ on a non-negligible fraction of inputs, then one can store these inputs with very few bits. This implies that π_n can be represented in $o(\log(2^n!))$ bits and one can use this to argue that the random permutations are hard relative to Breaker.

Another technique which seems a bit easier to apply for proving the hardness of random permutations relative to an oracle was developed by Haitner and Holenstein [12] following the work of Simon [23]. Informally, to show that an adversary with oracle access to Breaker and a random permutation π_n on $\{0,1\}^n$ can not find $\pi_n^{-1}(y)$ for $y \sim \mathcal{U}_n$, one modifies the permutation π by randomly selecting an $x^* \in \{0,1\}^n$ and mapping it to y. This new function is almost certainly not a permutation, but if one can show that the execution of the adversary changes after the modification with only negligible probability, then it implies that the adversary could not have found x^* as a pre-image, since after the change both $\pi_n^{-1}(y)$ and x^* are equally likely to be found as a pre-image.

Now we describe the above techniques in detail.

4.1.4.1. The Information Theoretic Method of Gennaro and Trevisan

In this section, we use a technique given by Gennaro and Trevisan [5] to prove the following information theoretic lemma which we then use to prove Lemma 4.1.3. In the following, we denote by π_{-n} an ensemble of permutations $\{\pi_n\}_{n\geq 1}$ where the permutation π_n is left undefined.

4.1.5 Lemma.
Let $\pi = \{\pi_n\}_{n\geq 1}$ be a function ensemble where $\pi_n : \{0,1\}^n \to \{0,1\}^n$ is a permutation. Let $F^{\pi,\mathsf{Breaker}^\pi}$ be a non-uniform oracle adversary which makes at most $q = q(n) \in \mathsf{poly}(n)$ oracle queries and inverts $\pi : \{0,1\}^n \to \{0,1\}^n$ on more than δ fraction of the inputs, then there exists $a \geq \frac{\delta 2^n}{q+1}$ such that given π_{-n} and $F^{\pi,\mathsf{Breaker}^\pi}$, the permutation π_n can be represented in at most

$$2\log\binom{2^n}{a} + a\,(n-k+\lceil\log(q)\rceil) + \log((2^n-a)!)$$

bits.

Before proving the above lemma, we will first prove how it implies Lemma 4.1.3.

▶ **Proof. (Lemma 4.1.3 from Lemma 4.1.5)** Lemma 4.1.5 implies that for every non-uniform oracle adversary F which makes at most $q = q(n) \in \mathsf{poly}(n)$ oracle queries, the fraction of permutations π_n for which

$$\mathbb{P}_{y\sim_{\mathcal{U}}\{0,1\}^n}[F^{\mathsf{Breaker},\pi}(1^n,y) = \pi_n^{-1}(y)] \geq \frac{\varepsilon}{2^{k/4}}$$

is at most

$$\frac{\binom{N}{a}^2 (N-a)!\; 2^{a(n-k+\log q)}}{N!} = \frac{\binom{N}{a}\; 2^{a(n-k+\log q)}}{a!}$$

where $N = 2^n$, $a \geq \frac{\varepsilon}{2^{k/4}}\frac{2^n}{q+1}$, $\varepsilon = \varepsilon(n) \in \mathsf{poly}^{-1}(n)$ and $k = \omega(\log n)$.

Using inequalities $a! \geq (a/e)^a$ and $\binom{N}{a} \leq (Ne/a)^a$, the expression above can be upper bounded by

$$\left(\frac{Ne^2}{a^2}\right)^a 2^{a(n-k+\lceil \log q \rceil)} \leq \left(\frac{e^2 \mathsf{poly}(n)}{2^{k/2}}\right)^a.$$

Since $k = \omega(\log n)$ the above is at most $2^{-n/2}$ for large enough n. Therefore, letting Π_n denote the distribution induced by random permutations on $\{0,1\}^n$, we have

$$\mathbb{P}_{\pi_n \sim \Pi_n, y \sim_U \{0,1\}^n}[F^{\mathsf{Breaker},\pi}(1^n, y) = \pi_n^{-1}(y)] \leq \frac{\varepsilon}{2^{k/4}} + 2^{-n/2}$$

which is a negligible function of n. ■

Now we prove Lemma 4.1.5.

► **Proof. (Lemma 4.1.5)** Consider the set I of bitstrings $y \in \{0,1\}^n$ on which $F^{\mathsf{Breaker},\pi}$ inverts π_n successfully. Note that by our assumptions $|I| \geq \varepsilon 2^n$. We want to argue that there exists a subset $Y \subseteq I$ of size at least $\frac{|I|}{q+1}$ such that given π_{-n}, $F^{\pi,\mathsf{Breaker}^\pi}$, sets Y and $\pi_n^{-1}(Y)$, the description of π on $\{0,1\}^n \setminus Y$ and an additional $n - k + \lceil \log q \rceil$ bits for each string in Y the value of π_n^{-1} is determined for all $y \in Y$.

Defining the set Y. We define Y be the following inefficient iterative process. Initially Y is empty. We pick the lexicographically smallest element y from I and insert it into Y. Then we follow the computation of $F^{\mathsf{Breaker},\pi}(1^n, y)$: let $x_1, \ldots, x_t$ be the queries which F makes to π_n. In addition, let $(1^{n_1}, w_1), \ldots, (1^{n_{t'}}, w_{t'})$ be the queries which F makes to Breaker and which do not get answered by $\perp$. Let $x'_1, \ldots, x'_{t'}$ denote the answer to these. We remove $\pi_n(x_1), \ldots, \pi_n(x_t)$ and $\pi_n(x'_1), \ldots, \pi_n(x'_{t'})$ from I (note that these might not be in the set I). Then, we remove the lexicographically smallest element from the remaining elements of I, put it in Y and iterate in the same manner as above until the set I is empty.

Note that at each iteration in the above process, we insert one element in Y and remove at most $t + t' \leq q + 1$ elements from I. It follows that when the above process terminates then $|Y| \geq \frac{|I|}{q+1}$. The description of Y and $\pi^{-1}(Y)$ both require $\log \binom{2^n}{|Y|}$ bits while the description of π on $\{0,1\}^n \setminus Y$ requires at most $\log((2^n - |Y|))$ bits.

We need to store additional bits for each element in $y \in Y$ to reconstruct π_n. This is done as follows. During the execution of the above process with y, if F makes a query to Breaker with input $(1^n, w)$ such that $w = (y, \pi_n^{-1}(y)|_{\{1,\ldots,k\}})$ (observe that in this case Breaker returns $\pi_n^{-1}(y)$ as answer), then we store the index of the current query among all the q queries that F makes. This takes $\lceil \log q \rceil$ bits. Furthermore, we also store the last $n - k$ bits of $\pi_n^{-1}(y)$. Overall, we store at most $n - k + \lceil \log q \rceil$ bits for each element in $y \in Y$.

Thus, the total number of bits we have stored is at most

$$2 \log \binom{2^n}{|Y|} + |Y|\,(n - k + \lceil \log(q) \rceil) + \log((2^n - |Y|)!).$$

We will next show that they suffice to reconstruct π_n completely.

Reconstructing π_n. We now claim that given the description of sets Y, $\pi_n^{-1}(Y)$, an additional $n-k+\lceil \log q \rceil$ bits stored as above for each string in Y and the description of π_n on $\{0,1\}^n \setminus Y$, the value of π_n^{-1} is determined for all $y \in Y$. For this, we will show how to reconstruct the pre-images for $y \in Y$ from the above information. We pick the lexicographically smallest element $y \in Y$ whose pre-image is not known and simulate the execution of $F^{\mathsf{Breaker}^{\pi},\pi}(1^n, y)$. We claim that we can correctly simulate the answer to all π_n queries with the above information. Note that if the simulation behaves correctly then this computation gives the value of $\pi_n^{-1}(y)$.

First note that any queries to π_m for $m \neq n$ can be answered as their values are explicitly known. Next consider all the π_n queries which the adversary F makes directly to π_n. By the definition of the set Y, either these queries are made with elements not in $\pi_n^{-1}(Y)$, or with strings x' such that $\pi(x')$ precedes y in the lexicographic order, or otherwise, the query is made with $\pi_n^{-1}(y)$ itself. In the first case the value is already known, in the second one, the value was already reconstructed since we are proceeding in the lexicographic order for reconstruction and in the final case, we already have the desired value $\pi_n^{-1}(y)$. In all cases we have the information to proceed with the correct simulation.

Now let us look at the queries which $\mathsf{Breaker}^{\pi}$ makes to π_n during the simulation. Note that if the Breaker queries π_n with x, then one of the following cases occurs :

- $x \in \{0,1\}^n \setminus \pi_n^{-1}(Y)$. In this case the value is given explicitly.
- $x \in \pi_n^{-1}(Y)$ but $\mathsf{Breaker}^{\pi}$ does not return x as an answer. In this case the value $\pi_n(x)$ might not be known but this does not affect the simulation since Breaker can continue with the next query to π_n.
- $x \in \pi_n^{-1}(Y)$ and $\pi_n(x)$ precedes y in the lexicographic order. In this case the required value was already reconstructed.
- $x = \pi_n^{-1}(y)$ and Breaker returns x as an answer. In this case, the value is not yet known but we can not stop and claim that we know the answer (as we did with the direct π_n queries of F), since the default behavior of the simulation of Breaker queries to π_n is to assume that x is not the correct answer for this query and continue trying the next bitstrings in sequence. In this case the simulation would be clearly wrong.

  This is where we utilize the $n-k+\lceil \log q \rceil$ extra bits we stored for each $y \in Y$. The first $\lceil \log q \rceil$ bits identifies that the current query is problematic and we need to use the extra bits we stored to reconstruct. Let $(1^n, w)$ where $w \in \{0,1\}^{n+k}$ be the input to Breaker. Then in this case we know the first k bits of x since $w|_{\{n+1,\ldots,n+k\}} = x|_{\{1,\ldots,k\}}$ and we already have the last $n-k$ bits stored up. So we have enough information to reconstruct $\pi_n^{-1}(y)$. ■

4.1.4.2. The Technique of Haitner and Holenstein

We now use the technique of Haitner and Holenstein to prove Lemma 4.1.3. Let $F^{\mathsf{Breaker}^{\pi},\pi}$ be a probabilistic polynomial time adversary with oracle access to Breaker and π. The adversary gets as input $(1^n, y)$ where $y \in \{0,1\}^n$ and tries to find $\pi_n^{-1}(y)$. Consider how the execution of the adversary changes if we perturb π_n by randomly choosing an $x^* \in \{0,1\}^n$ and mapping it to y. Lets denote

this function by $\pi_n^{x^*\to y}$ and use $\pi^{x^*\to y}$ to denote the ensemble generated from π by replacing π_n with $\pi_n^{x^*\to y}$. Though with high probability $\pi_n^{x^*\to y}$ is not a permutation, one expects that only with a very small probability there will be a change in the execution of the adversary when it uses the oracles $\mathsf{Breaker}^{\pi^{x^*\to y}}$ and $\pi^{x^*\to y}$ instead of $\mathsf{Breaker}^{\pi}$ and π.

So, if one can formally prove that the execution changes with a negligible probability $\varepsilon(n)$, then one can apply the same argument to the permutation where the pre-images of $\pi_n(x^*)$ and y are interchanged (now considering $\pi_n^{x^*\to y}$ as the modified version of the permutation). Let us denote this permutation by $\pi_{*,n}$ and the corresponding ensemble generated from π by replacing π_n with $\pi_{*,n}$ by π_*. The above implies that the adversary outputs the same answer as the pre-image of y for both π and $\pi_{*,n}$ while y actually has different pre-images under them. It follows that the probability that the adversary $F^{\mathsf{Breaker}^{\pi},\pi}$ inverts y is negligible for all $y \in \{0,1\}^n$.

We now formalize the above.

4.1.6 Lemma. (π remains hard)
There exists a negligible function $\varepsilon(n)$ such that for every $y \in \{0,1\}^n$ the probability (over the choice of x^*) that the execution of $F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)$ differs from the execution of $F^{\pi^{x^*\to y},\mathsf{Breaker}^{\pi^{x^*\to y}}}(1^n,y)$ is at most $\varepsilon(n)$.

▶ **Proof.** We will only consider the case when $x^* \neq \pi_n^{-1}(y)$ since otherwise the statement is trivial.

Consider the execution of $F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)$ and $F^{\pi^{x^*\to y},\mathsf{Breaker}^{\pi^{x^*\to y}}}(1^n,y)$ in parallel. $F^{\pi,\mathsf{Breaker}^{\pi}}$ makes queries to π and $\mathsf{Breaker}^{\pi}$ and computes an answer, while $F^{\pi^{x^*\to y},\mathsf{Breaker}^{\pi^{x^*\to y}}}$ queries $\pi^{x^*\to y}$ and $\mathsf{Breaker}^{\pi^{x^*\to y}}$. Since F is deterministic, the only way the execution can differ is when a particular query to π (or $\mathsf{Breaker}^{\pi}$) returns a different answer than the same query to $\pi^{x^*\to y}$ (or $\mathsf{Breaker}^{\pi^{x^*\to y}}$).

Let us now bound the probability that this happens for a fixed query assuming all the previous queries returned the same answer in both executions.

First note that, all queries made by F to π (or $\pi^{x^*\to y}$) or $\mathsf{Breaker}^{\pi}$ (or $\mathsf{Breaker}^{\pi^{x^*\to y}}$) for a security parameter other than 1^n return the same answer in both cases. So let us turn our attention to queries made only with security parameter 1^n. In this case, the answers to one of the queries to π versus $\pi^{x^*\to y}$ are different only when the adversary queries with x^*. The probability that this happens for a particular query is at most 2^{-n}.

Next we upper bound the probability that any fixed Breaker query leads to a change in the execution of the adversary. Notice that the change might occur only in one of the following two cases: either the input to Breaker is $(1^n,(\pi(x^*),z))$ for some $z \in \{0,1\}^k$. Or the input is $(1^n,(y,\pi^{-1}(y)_{[k]}))$ and $x^*|_{\{1,\ldots,k\}} = \pi_n^{-1}(y)|_{\{1,\ldots,k\}}$ holds. Since x^* is chosen independently and uniformly at random, the probability that the first case occurs is at most 2^{-n}. The second event occurs with probability at most 2^{-k}.

Considering all queries to π and Breaker simultaneously and using the union bound to bound the probability that one of them is the first one where the execution differs, we get that the probability that the execution of $F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)$ differs from the execution of $F^{\pi^{x^*\to y},\mathsf{Breaker}^{\pi^{x^*\to y}}}(1^n,y)$ is at most $\mathsf{poly}(n)2^{-k}$ which is negligible since $k=\omega(\log n)$. ■

The proof of Lemma 4.1.3 easily follows from the above lemma.

▶ **Proof. (Lemma 4.1.3)** Lemma 4.1.6 implies that for all $y \in \{0,1\}^n$ the probability that the execution of $F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)$ differs from the execution of $F^{\pi^{x^*\to y},\mathsf{Breaker}^{\pi^{x^*\to y}}}(1^n,y)$ is at most a negligible function $\varepsilon(n)$. The same holds for the execution of $F^{\pi^{x^*\to y},\mathsf{Breaker}^{\pi^{x^*\to y}}}(1^n,y)$ versus the execution of $F^{\pi_*,\mathsf{Breaker}^{\pi_*}}(1^n,y)$. In particular, we have that for all $y \in \{0,1\}^n$ $F^{\pi,\mathsf{Breaker}^{\pi}}(y)$ returns the same answer as as $F^{\pi_*,\mathsf{Breaker}^{\pi_*}}(y)$ with probability at least $1-2\varepsilon(n)$ over the choice of x^*.

Let Π_n denote the distribution induced by random permutations over $\{0,1\}^n$. If $\pi_n \sim \Pi_n$, then we have that for all $y \in \{0,1\}^n$

$$\begin{aligned}\mathbb{P}_{\pi_n\sim\Pi_n,x^*\sim\{0,1\}^n}[F^{\pi_*,\mathsf{Breaker}^{\pi_*}}(1^n,y)=\pi_*^{-1}(y)] \leq & \mathbb{P}_{\pi_n,x^*}[F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)=F^{\pi_*,\mathsf{Breaker}^{\pi_*}}(1^n,y)=\pi_*^{-1}(y)]\\ &+\mathbb{P}_{\pi_n,x^*}[F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)\neq F^{\pi_*,\mathsf{Breaker}^{\pi_*}}(1^n,y)]\\ \leq & \mathbb{P}_{\pi_n,x^*}[F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)=\pi_{*,n}^{-1}(y)]+2\varepsilon(n).\end{aligned}$$

Note that $\pi_{*,n}^{-1}(y)=x^*$ which is chosen independently at random from the execution of $F^{\pi,\mathsf{Breaker}^{\pi}}(1^n,y)$, so the first term above on the right hand side is at most 2^{-n}. The probability in the left hand side above can be written as $\mathbb{P}_{\pi_{*,n}\sim\Pi_n}[F^{\pi_*,\mathsf{Breaker}^{\pi_*}}(1^n,y)=\pi_{*,n}^{-1}(y)]$ since when $\pi_n \sim \Pi_n$, then $\pi_{*,n}$ is also distributed according to Π_n. The statement of the lemma follows. ■

4.1.5 Existence of Hard Permutations relative to Breaker

In this section we use the Borel-Cantelli Lemma and Lemma 4.1.3 to derive Lemma 4.1.4.

▶ **Proof. (Lemma 4.1.4)** For any PPT adversary F, let $\mathbf{E}_{n,F}$ denote the event that $\mathbb{P}_{y\sim\mathcal{U}_n}[F^{\mathsf{Breaker},\pi}(1^n,y)=\pi_n^{-1}(y)] \geq n^2\varepsilon(n)$ when π_n is drawn from Π_n where $\varepsilon(n)$ is the negligible function from Lemma 4.1.3. Markov's inequality then implies that for large enough n, the probability that the event $\mathbf{E}_{n,F}$ happens is at most $1/n^2$. Let $\mathbf{E}_{\infty,F}$ denote the event a permutation ensemble $\{\pi_n\}_{n\geq 1}$ is picked such that for infinitely many n, the event $\mathbf{E}_{n,F}$ happens. Since $\sum_{n\geq 1}\frac{1}{n^2}$, converges the Borel-Cantelli Lemma implies that $\mathbb{P}_{\{\pi_n\}_{n\geq 1}}[\mathbf{E}_{\infty,F}]=0$.

Let Ω_F be the set of permutation ensembles for which the event $\mathbf{E}_{\infty,F}$ occurs. Ω_F has measure 0 for any F (i.e. probability zero that $\mathbf{E}_{\infty,F}$ occurs). As there are countably many PPT adversaries F, the set $\Omega=\cup_F\Omega_F$ also has measure 0. In particular, this means that the probability over the choice of permutation ensembles that the event "$\vee_F\mathbf{E}_{\infty,F}$" happens is zero. ■

CHAPTER 5

Lower Bounds

Given any one-way function $\{f_n\}_{n\geq 1}$ where $f_n : \{0,1\}^n \to \{0,1\}^n$, let us consider non-adaptive black-box PRG constructions of the form $g(x_1,\ldots,x_t,f(x_1),\ldots,f(x_t))$ for $t \geq 1$. In this chapter we seek to show that for small values of t, it is impossible to give black-box PRG constructions of the above form. The main results which we prove are the following :

- For $t = 1$, we show that a black-box PRG construction of the above form is impossible.
- For $t > 1$, we prove that if g satisfies a combinatorial condition (which depends on t), then $g(x_1,\ldots,x_t,f(x_1),\ldots,f(x_t))$ can not be a black-box PRG construction. We conjecture that for $t \leq n/\log^2 n$, all function ensembles g of the form $g(x_1,\ldots,x_t,f(x_1),\ldots,f(x_t))$ satisfy this condition and under this conjecture we obtain a non-trivial $\Omega(n^2/\log^2 n)$ lower bound on the seed length of such non-adaptive black-box PRG constructions.

We start in the next section by proving the impossibility result for $t = 1$. In Section 5.2, we discuss the case of $t = 2$ and state the combinatorial conjecture. In Section 5.3, we generalize the above for cases where $t > 2$.

5.1 Impossibility of black-box PRG constructions of the form $g(x, f(x))$

We want to show that there does not exist any black-box construction of the form $\{g_n\}_{n\geq 1}, g_n : \{0,1\}^{2n} \to \{0,1\}^{n+1}$ such that when $x \sim \mathcal{U}_n$, $g_n(x,f_n(x))$ is pseudorandom for every one-way function $\{f_n\}_{n\geq 1}, f_n : \{0,1\}^n \to \{0,1\}^n$. More formally we prove the following theorem. To emphasize that this is a black-box impossibility result we state the theorem in terms of functions with access to oracles.

▶ **Theorem. 5.1.1 (Impossibility of Black-box PRG construction with single query)** *There does not exist any oracle function ensemble $\{g_n\}_{n\geq 1}, g^{(\cdot)} : \{0,1\}^n \to \{0,1\}^{n+1}$ such that the following holds for every function ensemble $\{f_n\}_{n\geq 1}, f : \{0,1\}^n \to \{0,1\}^n$ for infinitely many n.*

- *For any* $x \in \{0,1\}^n$, $g_n^{f_n}(x) = g_n(x, f_n(x))$.
- *If* f *is secure, then* $g^f(\mathcal{U}_n)$ *is computationally indistinguishable from* $\mathcal{U}_{n+1}$ *with a black-box proof of security.*

Following the techniques from Chapter 3, to prove the above it suffices to show that there exists an appropriate separation oracle O and a one-way function f such that f is secure relative to O while g is not. Note that we can not take f to be a one-way permutation because of Theorem 2.3.6.

Consider first the following natural candidate for separation oracle (Algorithm 2) à la Chapter 3. The oracle when given an input $(1^n, w)$ where $w \in \{0,1\}^{n+1}$, goes through all $x \in \{0,1\}^n$, applies f_n on x, and simply checks if $g_n(x, f_n(x)) = w$.

Algorithm 2: $\mathsf{SimpleBreaker}^{f,g}(1^n, w)$:

```
for all x ∈ {0,1}^n do
    if g_n(x, f_n(x)) = w then
        return x;
return ⊥
```

The oracle described above does not give us a black-box separation. In fact, it is too powerful and can be used to break any one-way function f if we choose a suitable function g, *e.g.* $g(x,y) = (y,1)$. The inverter then gets $(1^n, f_n(x))$ as input for $x \sim \mathcal{U}_n$ and simply queries SimpleBreaker with $(1^n, (f_n(x), 1))$ and gets a pre-image of x. However, the above should not come as too much of a surprise since SimpleBreaker does not make any distinction between one-way permutations and degenerate one-way functions which as mentioned above is crucial.

Lets delve a bit into thinking how can we fix the above SimpleBreaker. Consider any k-degenerate regular one-way function ensemble f where $f_n : \{0,1\}^n \to \{0,1\}^n$. Given $y = f_n(x)$ where $x \sim \mathcal{U}_n$, there are 2^k pre-images $f_n^{-1}(y)$ any of which are acceptable as an answer given by an adversary. Or to interpret differently, the adversary is only interested in finding $n-k$ bits of information about x. Now, given any candidate PRG construction g where $g_n : \{0,1\}^{2n} \to \{0,1\}^{n+1}$, if SimpleBreaker on security parameter n is queried with a w such that "knowing" y, w and g already determines that the set $S = \{x \in \{0,1\}^n \mid g_n(x,y) = w\}$ has much fewer than 2^{n-k} elements, then such queries are fine since w (and y) determine much more than k bits of entropy of x and so any adversary which does not already know significant information about a pre-image of y could not have produced w. On the other hand, if $|S|$ is much larger than 2^{n-k}, then the adversary might produce a suitable w to query just using y and few random bits and get the pre-image by querying SimpleBreaker. We would like to prevent the latter as it can be used by an adversary to invert one-way functions.

However, if we just forbid the above then the oracle becomes too weak as we are not able to detect those $w \in \mathsf{Im}(g)$ as pseudorandom which have too much probability mass. So, we can not break extremely degenerate PRG candidates, like $g_n(x,y) = 0^{n+1}$, for example, because they will fail the condition for most $w \in \mathsf{Im}(g_n)$. In such cases though, we can just check whether the probability mass attached to $w \in \{0,1\}^{n+1}$ is much larger than 2^{-n}. As we will show now that the above

two tweaks are sufficient for our purposes. We show how to analyze g and based on that select a suitable function ensemble f where $f_n : \{0,1\}^n \to \{0,1\}^n$ such that $g_n(x, f_n(x))$ lands in one of the aforementioned cases with high probability for infinitely many n.

We now formulate the above ideas more precisely. Let $w \in \{0,1\}^{n+1}$ and $y \in \{0,1\}^n$ be bitstrings. Let X be an independent random variable distributed according to $\mathcal{U}_n$. We will take $H_0(X \mid g_n(X,y) = w)$ as a measure of how much of the entropy of x is determined given w and y when x is drawn from $\mathcal{U}_n$. Then we proceed as discussed above. Note that $H_0(X \mid g_n(X,y) = w) = k$ is equivalent to saying that the size of the set $\{x \in \{0,1\}^n \mid g_n(x,y) = w\}$ is 2^k. We will often use both these notions interchangeably.

5.1.1 The Separation Oracle Breaker

Let us say that we have a candidate PRG ensemble g where $g_n : \{0,1\}^{2n} \to \{0,1\}^{n+1}$ for which we want to give a separation oracle. We first define a family of separation oracle Breaker parametrized by some $\varepsilon = \varepsilon(n)$ where $\omega(\log n/n) = \varepsilon < 1/1000$ and a list of integers $\{k_n\}_{n\geq 1}$ where $k_n \in [\varepsilon n, n - \varepsilon n]$. Later we will show that a particular oracle from this family achieves separation for the ensemble g.

The separation oracle with parameter ε and k (denoted as $\mathsf{Breaker}_{k,\varepsilon}$) works as follows. If y and w are such that $H_0(X \mid g_n(X,y) = w) \leq n - k_n - \varepsilon n$ for a random variable $X \sim \mathcal{U}_n$, then w extracts too much information about x and so to produce the right w to query Breaker, the adversary must already know significant information about x. So, in this case, Breaker returns the x which matches. On the other hand if $H_0(X \mid g_n(X,y) = w) \geq n - k_n + \varepsilon n$ for some $y \in \{0,1\}^n$, then the Breaker just outputs Pseudorandom. It will turn out that these are the elements which have a large probability mass. For better exposition we label these blocks of code of Breaker as $\mathsf{Breaker}+^{g,S}_{k,\varepsilon}(1^n, w)$ and $\mathsf{Breaker}-^{f,g}_{k,\varepsilon}(1^n, w)$. A formal description of Breaker is given in Algorithm 3. Note that Breaker also has oracle access to an ensemble $\{S_n\}_{n\geq 1}$ where $S_n \subseteq \{0,1\}^n$.

Remark about notation. In the reminder of this chapter, we will try to omit parameters and oracles from the description of Breaker if they are clear from the context.

5.1.2 The Hard Distribution of Functions

In this section we will show how to choose a family of functions f_n which will be approximately k_n-degenerate for some $k_n \in [\varepsilon n, n - \varepsilon n]$ with the property that for some function ensemble f where $f_n : \{0,1\}^n \to \{0,1\}^n$ is drawn from the family, the Breaker with parameter $k := \{k_n\}_{n\geq 1}$ and ε will allow us to distinguish the output of $g_n(x, f_n(x))$ from $\mathcal{U}_{n+1}$ with non-negligible probability for infinitely many n, but f remains hard relative to the same oracle.

For each $n \geq 1$, first we choose the degeneracy k_n of the function f_n depending on g_n in the following manner. Take k_n to be the largest integer in the interval $[\varepsilon n, n - \varepsilon n]$ such that

$$\mathbb{P}_{x\sim\mathcal{U}_n, y\sim\mathcal{U}_n}\left[\, H_0(X \mid g_n(X,y) = g_n(x,y)) \in (k_n - \varepsilon n, k_n + \varepsilon n]\, \right] \leq 2\varepsilon. \tag{5.1}$$

Note that such a k_n must always exist since otherwise we could divide the interval $[0, n]$ into at least $1/2\varepsilon$ disjoint intervals ($[0, 2\varepsilon n]$, $(2\varepsilon n, 4\varepsilon n]$ and so on) such that the probability that $g_n(x,y)$ lands in each such interval is strictly greater than $1/2\varepsilon$, which violates the axiom of total probability.

Algorithm 3: $\mathsf{Breaker}^{f,g,S}_{k,\varepsilon}(1^n, w)$:

```
// X ~ U_n in the following;
// the begin statements in the following just demarcate a block of code;
1  begin // Breaker+^{g,S}_{k,ε}(1^n, w)
2      for all y ∈ S_n do
3          if H_0(X | g_n(X, y) = w) ≥ n − k_n + εn      (★) then
4              return Pseudorandom;
5  begin // Breaker−^{f,g}_{k,ε}(1^n, w)
6      for all y ∈ {0,1}^n do
7          if H_0(X | g_n(X, y) = w) ≤ n − k_n − εn      (★★) then
8              for all x ∈ {0,1}^n do
9                  if g_n(x, y) = w and f_n(x) = y then
10                     return x;
11 return ⊥
```

(We assume that $1/2\varepsilon$ and εn are integers. Otherwise, we have to take floors and ceilings everywhere which apart from being a mild annoyance does not affect the following parts of the proof.)

Moreover, by an averaging argument (5.1) also implies that there exists a subset $S'_n \in \{0,1\}^n$, $|S'_n| \geq 2^{n-1}$ such that for all $y \in S'_n$

$$\mathbb{P}_{x\sim\mathcal{U}_n}\left[\, H_0(X \mid g_n(X,y) = g_n(x,y)) \in (k_n - \varepsilon n, k_n + \varepsilon n]\, \right] \leq 4\varepsilon. \tag{5.2}$$

We take any arbitrary subset $S_n \subseteq S'_n$ of size $|S_n| = 2^{n-k_n}$ to be a superset of the image of f_n.

With the above in hand, consider the following distribution over functions from $\{0,1\}^n$ to $\{0,1\}^n$.

5.1.2 Definition (Hard Distribution of Functions $\mathcal{F}_{n,S_n}$)
The distribution of functions $\mathcal{F}_{n,S_n}$ is the distribution induced by uniform random functions from $\{0,1\}^n$ to S_n.

The ensemble of functions f which we will work with is the one where $f_n \sim \mathcal{F}_{n,S_n}$. We remark that the distribution $\mathcal{F}_{n,S_n}$ depends on the parameter ε.

For security parameter 1^n, choosing f_n from the above distribution will allow us to ensure that with high probability over the choice of f_n almost every $w \in \mathsf{Im}(g_n)$ is such that we can use $\mathsf{Breaker}_{k,\varepsilon}$ to distinguish it from a uniform random bitstring of length $n+1$.

5.1.3 A Distinguisher for $g(x, f(x))$

Let $f = \{f_n\}_{n\geq 1}$ be an ensemble where $f_n \sim \mathcal{F}_{n,S_n}$, $k = \{k_n\}_{n\geq 1}$ be the list of k_n's defined by (5.1) and $S = \{S_n\}_{n\geq 1}$. We now show that $\mathsf{Breaker}^{f,g,S}_{k,\varepsilon}$ with parameters k and ε allows us to build a distinguisher for $g(x, f(x))$.

Consider the following distinguisher $G^{\mathsf{Breaker}_{k,\varepsilon},f}(1^n, w)$: Output 1 if $\mathsf{Breaker}_{k,\varepsilon}(1^n, w)$ does not return $\perp$. This simple distinguisher allows us to distinguish $g_n(\mathcal{U}_n, f_n(\mathcal{U}_n))$ from $\mathcal{U}_{n+1}$ with non-negligible probability for infinitely many n. The main idea in proving the following lemma about the distinguishing advantage of G is to show that due to our choice of k, f and S, for most $w \in \mathsf{Im}(g_n)$ and any $y \in \mathsf{Im}(f_n)$, we either have $H_0(X \mid g_n(X, y) = w) \geq n - k_n + \varepsilon n$ in which case Breaker outputs Pseudorandom or $H_0(X \mid g_n(X, y) = w) \leq n - k_n - \varepsilon n$ where Breaker inverts g_n without violating the one-wayness of f.

5.1.3 Lemma.
The following holds for every $\omega(\log n/n) = \varepsilon < 1/1000$: let $f = \{f_m\}_{m\geq 1}$ be a function ensemble where $f_m : \{0,1\}^m \to \{0,1\}^m$ is a function drawn from the distribution $\mathcal{F}_{m,S_m}$ defined in Definition 5.1.2. Then for large enough n, with probability at least $1 - e^{-n}$ over the choice of the function f_n, the distinguishing advantage for the above distinguisher $G^{\mathsf{Breaker}_{k,\varepsilon},f}$ on security parameter 1^n is at least $\frac{1}{2} - 16\varepsilon - 2^{-\varepsilon n}$, i.e.

$$\left| \mathbb{P}[G^{\mathsf{Breaker}_{k,\varepsilon},f}(1^n, g_n^{f_n}(\mathcal{U}_n)) = 1] - \mathbb{P}[G^{\mathsf{Breaker}_{k,\varepsilon},f}(1^n, \mathcal{U}_{n+1}) = 1] \right| \geq \frac{1}{2} - 16\varepsilon - 2^{-\varepsilon n}.$$

▶ **Proof.** Let $w = g_n(x, y)$ where $x \sim \mathcal{U}_n$ and $y = f_n(x)$. Then the distinguisher does not output 1 if $H_0(X \mid g_n(X, y) = w) \in (n - k_n - \varepsilon n, n - k_n + \varepsilon n)$. Recall that we chose k_n and $\mathsf{Im}(f_n) \subseteq S_n$ such that for all $y \in \mathsf{Im}(f_n)$

$$\mathbb{P}_{x\sim\mathcal{U}_n}[H_0(X \mid g_n(X, y) = g_n(x, y)) \in (n - k_n - \varepsilon n, n - k_n + \varepsilon n] \,] \leq 4\varepsilon. \tag{5.3}$$

Now we claim that the above implies that for n large enough with probability at least $1 - e^{-n}$ over the choice of f_n we have the following :

$$\mathbb{P}_{x\sim\mathcal{U}_n}[H_0(X \mid g_n(X, f_n(x)) = g_n(x, f_n(x))) \in (n - k_n - \varepsilon n, n - k_n + \varepsilon n] \,] \leq 16\varepsilon. \tag{5.4}$$

In this case, it follows that the distinguisher $G^{\mathsf{Breaker}_{k,\varepsilon},f}$ outputs 1 on $(1^n, g_n^{f_n}(\mathcal{U}_n))$ with probability at least $1 - 16\varepsilon$.

To see the above claim, note that (5.3) implies that for each $y \in \mathsf{Im}(f_n)$ at most $4\varepsilon 2^n$ strings $x \in \{0,1\}^n$ are such that $H_0(X \mid g_n(X, y) = g_n(x, y)) \in (n - k_n - \varepsilon n, n - k_n + \varepsilon n]$. Lets say that the above x are *bad* for y. Define $X_{f_n,y}$ to be the set of pre-images of y under f_n and let $X^*_{f_n,Y}$ denote the set of pre-images of y which are bad.

Note that due to Chernoff bounds, for a fixed $y \in \mathsf{Im}(f_n)$ with probability at least $1 - e^{-n^2}$ over the choice of f_n, we have $2^{k_n-1} \leq |X_{f_n,y}| \leq 2^{k_n+1}$ (see Lemma 2.4.2). Also a similar application of Chernoff bounds gives us that with probability at least $1 - e^{-n^2}$, $4\varepsilon \cdot 2^{k_n-1} \leq |X^*_{f_n,y}| \leq 4\varepsilon \cdot 2^{k_n+1}$. This together with a union over all $y \in \mathsf{Im}(f_n)$ implies that the following holds with probability at least $1 - e^{-n}$ (over the choice of f_n) for all $y \in \mathsf{Im}(f_n)$,

$$\mathbb{P}_{x\sim\mathcal{U}_n}[H_0(X \mid g_n(X, y) = g_n(x, y) \in (n - k_n - \varepsilon n, n - k_n + \varepsilon n] \mid f_n(x) = y \,] \leq 16\varepsilon.$$

Let f_n be a function which satisfies the above (this happens with probability at least $1-e^{-n}$ when $f_n \sim \mathcal{F}_{n,S_n}$), then we can write

$$\begin{aligned}
&\mathbb{P}_{x\sim\mathcal{U}_n}[H_0(X \mid g_n(X,f(x)) = g_n(x,f_n(x))) \in (n-k_n-\varepsilon n, n-k_n+\varepsilon n]\ ] \\
&\quad = \sum_{y\in \mathsf{Im}(f_n)} \mathbb{P}_{x\sim\mathcal{U}_n}[H_0(X \mid g_n(X,y) = g_n(x,y)) \in (n-k_n-\varepsilon n, n-k_n+\varepsilon n] \mid f_n(x)=y\ ] P[f_n(x)=y] \\
&\quad \leq 16\varepsilon \cdot \sum_{y\in \mathsf{Im}(f_n)} P[f_n(x)=y] = 16\varepsilon,
\end{aligned}$$

which proves the required claim (5.4).

On the other hand if $w \sim \mathcal{U}_{n+1}$, then we claim that the distinguisher will output 1 with probability at most $\frac{1}{2} + 2^{-\varepsilon n}$. To see this first note that at most $\frac{1}{2}$ fraction of the $w \in \{0,1\}^{n+1}$ are such that Breaker$-_{k,\varepsilon}$ is invoked and it returns an answer. This is so because Breaker$-_{k,\varepsilon}$ checks whether w is in the set $\{g_n(x,f_n(x)) \mid x \in \{0,1\}^n\}$ and this set has size at most 2^n.

For the cases where Breaker$+_{k,\varepsilon}$ is invoked and returns Pseudorandom as the answer, define the set $W \subseteq \mathsf{Im}(g_n)$ as follows:

$$W = \{w \in \mathsf{Im}(g_n) \mid \exists y \in S_n : H_0(X \mid g_n(X,y) = w) \geq n - k_n + \varepsilon n\}.$$

Note that W contains exactly those $w \in \{0,1\}^{n+1}$ on which Breaker$+_{k,\varepsilon}$ outputs Pseudorandom. We now claim that for each $y \in S_n$, there are at most $2^{k_n-\varepsilon n}$ many strings $w \in W$ such that $H_0(X \mid g_n(X,y) = w) \geq n - k_n + \varepsilon n$. To see this recall that the definition of max-entropy implies that if $H_0(X \mid g_n(X,y) = w) \geq n - k_n + \varepsilon n$, then the set $\{x \in \{0,1\}^n \mid g_n(x,y) = w\}$ is of size at least $2^{n-k_n+\varepsilon n}$. This implies that for each $y \in S_n$, the number of strings $w \in W$ satisfying $H_0(X \mid g_n(X,y) = w) \geq n - k_n + \varepsilon n$ can be at most $\frac{\mathsf{Supp(X)}}{2^{n-k_n+\varepsilon n}} = \frac{2^n}{2^{n-k_n+\varepsilon n}} = 2^{k_n-\varepsilon n}$.

Note that since $|S_n| = 2^{n-k_n}$, the above implies that $|W| \leq 2^{k_n-\varepsilon n}\ |S_n| \leq 2^{n-\varepsilon n}$. Hence, the probability that Breaker$+_{k,\varepsilon}$ outputs Pseudorandom is at most $|W|/2^{n+1} \leq 2^{-\varepsilon n}$.

The claim about the distinguishing advantage follows. ■

Applying the Borel-Cantelli Lemma to the above we get the following.

5.1.4 Lemma.
For every $\omega(\log n/n) = \varepsilon < 1/1000$ the following holds : For any function ensemble $\{g_n\}_{n\geq 1}, g_n : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ there exists a non-negligible function $\delta(n)$ and a distinguisher $G^{\mathsf{Breaker}_{k,\varepsilon}}$ such that with probability 1 over the choice of function ensembles $\{f_n\}_{n\geq 1}$ where f_n is drawn as per the distribution $\mathcal{F}_{n,S_n}$, the following holds

$$\left|\mathbb{P}_{x\sim\mathcal{U}_n}[G^{\mathsf{Breaker}_{k,\varepsilon},f}(1^n, g_n(x,f_n(x))) = 1] - \mathbb{P}_{w\sim\mathcal{U}_{n+1}}[G^{\mathsf{Breaker}_{k,\varepsilon},f}(1^n,w) = 1]\right| \geq \delta(n).$$

5.1.4 f remains secure relative to Breaker

We now use the technique of Haitner and Holenstein [12] to show that a function drawn from the distribution $\mathcal{F}_n$ remains hard with respect to Breaker$_{k,\varepsilon}$. The formal statement is as follows.

5.1.5 Lemma.
For every $\omega(\log n/n) = \varepsilon < 1/1000$ the following holds : Let $f = \{f_n\}_{n\geq 1}$ be a function ensemble where $f : \{0,1\}^n \to \{0,1\}^n$ is a function distributed according to the distribution $\mathcal{F}_{n,S_n}$ defined in Definition 5.1.2 and let $F^{f,\mathsf{Breaker}_{k,\varepsilon}}(y)$ be any probabilistic polynomial time adversary that takes as input $y \in \{0,1\}^n$ and has oracle access to $\mathsf{Breaker}_{k,\varepsilon}$ and f. Then there exists a negligible function $\gamma(n)$ such that the following holds.

$$\mathbb{P}_{f_n\sim\mathcal{F}_{n,S_n},y\sim\mathcal{U}_n}[F^{f,\mathsf{Breaker}_{k,\varepsilon}}(1^n, y) \in f_n^{-1}(y)] \leq \gamma(n).$$

Note that in the above statement we can restrict our attention only to deterministic adversaries since any PPT adversary F which has advantage δ can be converted to a deterministic polynomial time adversary F' with advantage at least δ by fixing the best choice for internal random coins. So for the purpose of proving the above lemma, we will only take into consideration deterministic polynomial time adversaries.

To prove the lemma we proceed as in Chapter 4. Let $F^{f,\mathsf{Breaker}_{k,\varepsilon}}$ be a deterministic polynomial time adversary with oracle access to Breaker and f. The adversary gets as input $(1^n, y)$ where $y \in \{0,1\}^n$ and tries to find $f_n^{-1}(y)$. Consider how the execution of the adversary changes if we perturb f_n by randomly choosing an $x^* \in \{0,1\}^n$ and mapping it to y. Lets denote this new function by $f_{*,n}$ and denote by f_* the ensemble generated from f by replacing f_n by $f_{*,n}$. We show that only with a very small probability there will be a change in the execution of the adversary when it uses the oracles $\mathsf{Breaker}^{f_*}$ and f_* instead of $\mathsf{Breaker}^f$ and f.

5.1.6 Lemma. ($\mathcal{F}_{n,S_n}$ remains hard)
Fix any $\omega(\log n/n) = \varepsilon < 1/1000$. Then for every $y \in \{0,1\}^n$ and any polynomial time deterministic adversary $F^{f,\mathsf{Breaker}_{k,\varepsilon}}$, the probability over the choice of x^* that the execution of $F^{f,\mathsf{Breaker}^f_{k,\varepsilon}}(1^n, y)$ differs from the execution of $F^{f_*,\mathsf{Breaker}^{f_*}_{k,\varepsilon}}(1^n, y)$ is at most $\mathsf{poly}(n)2^{-k_n-\varepsilon n}$.

▶ **Proof.** We assume that $x^* \notin f^{-1}(y)$, since the statement holds trivially otherwise.

Consider the execution of $F^{f,\mathsf{Breaker}^f_{k,\varepsilon}}(1^n, y)$ and $F^{f_*,\mathsf{Breaker}^{f_*}_{k,\varepsilon}}(1^n, y)$ in parallel. $F^{f,\mathsf{Breaker}^f_{k,\varepsilon}}$ makes queries to f and $\mathsf{Breaker}_{k,\varepsilon}$ and computes an answer, while $F^{f_*,\mathsf{Breaker}^{f_*}_{k,\varepsilon}}$ queries f_* and $\mathsf{Breaker}^{f_*}_{k,\varepsilon}$. Since F is deterministic, the only way the execution can differ is when a particular query to f (or $\mathsf{Breaker}^f_{k,\varepsilon}$) returns a different answer than the same query to f_* (or $\mathsf{Breaker}^{f_*}_{k,\varepsilon}$).

Let us now bound the probability that this happens for a fixed query assuming all the previous queries returned the same answer in both executions.

First note that, all queries made by F to f (or f_*) or $\mathsf{Breaker}^f_{k,\varepsilon}$ (or $\mathsf{Breaker}^{f_*}_{k,\varepsilon}$) for a security parameter other than 1^n return the same answer in both cases. So let us turn our attention to queries made only with security parameter 1^n. In this case, the answers to one of the queries to f versus f_* are different only when the adversary queries with x^*. The probability that this happens for a particular query is at most 2^{-n}.

Next we upper bound the probability that any fixed $\mathsf{Breaker}_{k,\varepsilon}$ query is the first one where a difference in the answers returned by $\mathsf{Breaker}^f_{k,\varepsilon}$ and $\mathsf{Breaker}^{f_*}_{k,\varepsilon}$ occurs. First note that the execution of $\mathsf{Breaker}+_{k,\varepsilon}$ is not affected by the change so if the $\mathsf{Breaker}^f_{k,\varepsilon}$ returns Pseudorandom, then a query to $\mathsf{Breaker}^{f_*}_{k,\varepsilon}$ will also return the same.

So, the answers to $\mathsf{Breaker}_{k,\varepsilon}$ queries can be different in the two executions above only if one of the following happens :

- The adversary queries with some $w \in \{0,1\}^{n+1}$ and $\mathsf{Breaker}^{f}_{k,\varepsilon}$ outputs x^* while $\mathsf{Breaker}^{f_*}_{k,\varepsilon}$ outputs something else. Note that in this case the event $x^* = \mathsf{Breaker}^{f}_{k,\varepsilon}(1^n, w)$ happens and so the probability for this is at most 2^{-n}.
- The adversary queries with some $w \in \{0,1\}^{n+1}$ and $\mathsf{Breaker}^{f}_{k,\varepsilon}$ returns either some pre-image of y under f as the answer or $\perp$, while $\mathsf{Breaker}^{f_*}_{k,\varepsilon}$ outputs x^*. Note that this happens exactly if $H_0(X \mid g_n(X,y) = w) \leq n - k_n - \varepsilon n$ for $X \sim \mathcal{U}_n$ and $g_n(x^*, y) = w$. The first condition implies that the set $\{x \in \{0,1\}^n \mid g_n(x,y) = w\}$ has size at most $2^{n-k_n-\varepsilon n}$. Since x^* is chosen uniformly at random and independently the probability that it falls within this set is at most $2^{-k_n-\varepsilon n}$.

The proof is then completed by considering all the queries to f and $\mathsf{Breaker}_{k,\varepsilon}$ simultaneously and using the union bound to bound the probability that the one of them is the first one where the execution differs. ■

The proof of Lemma 5.1.5 easily follows from the above lemma.

▶ **Proof. (Lemma 5.1.5)** Let $\bar{x}$ be a fixed pre-image of y under f_n and $x^* \in \{0,1\}^n$ be picked uniformly at random. Define $f_{\dagger,n} : \{0,1\}^n \to \mathsf{Im}(f)$ to be the same as f_n except the images of $\bar{x}$ and x^* are interchanged. Let $f_\dagger$ denote the ensemble generated from f by replacing f_n by $f_{\dagger,n}$. Lemma 5.1.6 then implies that for all $y \in \{0,1\}^n$ the probability that the execution of $F^{f,\mathsf{Breaker}^{f}_{k,\varepsilon}}(1^n, y)$ differs from the execution of $F^{f_*,\mathsf{Breaker}^{f_*}_{k,\varepsilon}}(1^n, y)$ is at most $\mathsf{poly}(n)2^{-k_n-\varepsilon n}$. Note that if $f_n \sim \mathcal{F}_{n,S_n}$, then $f_{\dagger,n}$ is also distributed as $\mathcal{F}_{n,S_n}$. So the probability that the execution of $F^{f_\dagger,\mathsf{Breaker}^{f_\dagger}_{k,\varepsilon}}(1^n, y)$ differs from that of $F^{f_*,\mathsf{Breaker}^{f_*}_{k,\varepsilon}}(1^n, y)$ is also at most $\mathsf{poly}(n)2^{-k_n-\varepsilon n}$ since we can now consider $f_{*,n}$ as the modified version of $f_{\dagger,n}$. In particular, we have that for all $y \in \{0,1\}^n$ $F^{f,\mathsf{Breaker}^{f}_{k,\varepsilon}}(1^n, y)$ behaves identically to $F^{f_\dagger,\mathsf{Breaker}^{f_\dagger}_{k,\varepsilon}}(1^n, y)$ with probability at least $1 - 2\mathsf{poly}(n)2^{-k_n-\varepsilon n}$ over the choice of x^*. Denote the event that the execution of $F^{f_\dagger,\mathsf{Breaker}^{f_\dagger}_{k,\varepsilon}}(1^n, y)$ is the same as the execution of $F^{f,\mathsf{Breaker}_{k,\varepsilon}}(1^n, y)$ by $\mathbf{E}_{f,f_\dagger}$. Using the above we can write the following :

$$
\begin{aligned}
\mathbb{P}_{f,x^*\sim\mathcal{U}_n,F}[F^{f_\dagger,\mathsf{Breaker}^{f_\dagger}_{k,\varepsilon}}(1^n,y) \in {f_{\dagger,n}}^{-1}(y)] &\leq \mathbb{P}_{f,x^*,F}[F^{f_\dagger,\mathsf{Breaker}^{f_\dagger}_{k,\varepsilon}}(1^n,y) \in {f_{\dagger,n}}^{-1}(y) \mid \mathbf{E}_{f,f_\dagger}] \\
&\quad + \mathbb{P}_{f,x^*,F}[\neg\mathbf{E}_{f,f_\dagger}]) \\
&\leq \sum_{x \in {f_{\dagger,n}}^{-1}(y)} \mathbb{P}_{f,x^*,F}[F^{f,\mathsf{Breaker}^{f}_{k,\varepsilon}}(1^n,y) = x] + \mathsf{poly}(n)2^{-k_n-\varepsilon n} \\
&= \mathbb{P}_{f,x^*,F}[F^{f,\mathsf{Breaker}^{f}_{k,\varepsilon}}(1^n,y) = x^*] + \mathbb{P}_{f,F}[F^{f,\mathsf{Breaker}^{f}_{k,\varepsilon}}(1^n,y) \in f_n^{-1}(y)] \\
&\quad - \mathbb{P}_{f,F}[F^{f,\mathsf{Breaker}^{f}_{k,\varepsilon}}(1^n,y) = \bar{x}] + \mathsf{poly}(n)2^{-k_n+\varepsilon n}.
\end{aligned}
$$

Note that if $f \sim \mathcal{F}_{n,S_n}$ then $f_{\dagger,n}$ is also distributed as $\mathcal{F}_{n,S_n}$. So the term on the left hand side above equals $\mathbb{P}_{f,F}[F^{f,\mathsf{Breaker}^{f}_{k,\varepsilon}}(1^n,y) \in f_n^{-1}(y)]$. Also note that the first term on the right hand side equals 2^{-n} since x^* is picked independently of everything else. Hence previous calculations give us

$\mathbb{P}_f[F^{f,\mathsf{Breaker}^f_{k,\varepsilon}}(1^n, y) = \bar{x}] \leq 2^{-n} + 2^{-k_n+\varepsilon n}$. Since this holds for all pre-images $\bar{x} \in f_n^{-1}(y)$, and with probability at least $1 - e^{-n^2}$ the number of pre-images for every $y \in \mathsf{Im}(f_n)$ is at most 2^{k_n+1} due to Chernoff bounds (See Lemma 2.4.2), we have that $\mathbb{P}_{f,F}[F^{f,\mathsf{Breaker}^f_{k,\varepsilon}}(1^n, y) \in f_n^{-1}(y)] \leq \mathsf{poly}(n)2^{-\varepsilon n}$ which is negligible. ■

Remark : Note that using the technique of Gennaro and Trevisan, one can prove a lemma similar to Lemma 5.1.5 by working with a different distribution of functions which is also approximately k_n-regular for each $n \geq 1$, such as $h_{S_n}(\pi_n(x)|_{\{1,\ldots,n-k\}})$ where $h_{S_n} : \{0,1\}^n \to S_n$ is a universal hash function and π_n is a random permutation on $\{0,1\}^n$. The proof is very similar to the proof of Lemma 4.1.5 and so we omit it here.

Now we use the Borel-Cantelli Lemma and Lemma 5.1.5 to prove the following lemma which together with Lemma 5.1.4 proves Theorem 5.1.1.

5.1.7 Lemma.
For any $\omega(\log n/n) = \varepsilon < 1/1000$ the following holds : There exists a negligible function $\gamma'(n)$ such that with probability 1 over the choice of function ensembles $\{f_n\}_{n\geq 1}$ where f_n is distributed according to the distribution $\mathcal{F}_{n,S}$, the following holds for all probabilistic polynomial time adversaries $F^{f,\mathsf{Breaker}_{k,\varepsilon}}$ (with oracle access to $\mathsf{Breaker}^{f,g,S}_{k,\varepsilon}$ and f) and large enough n.

$$\mathbb{P}_{y\sim\mathcal{U}_n}[F^{f,\mathsf{Breaker}_{k,\varepsilon}}(1^n, y) \in f_n^{-1}(y)] \leq \gamma'(n).$$

▶ **Proof. (Lemma 5.1.7)** For any PPT adversary F, let $\mathbf{E}_{n,F}$ denote the event that $\mathbb{P}_{y\sim\mathcal{U}_n}[F^{f,\mathsf{Breaker}_{k,\varepsilon}}(1^n, y) \in f_n^{-1}(y)] \geq n^2\gamma(n)$ when f_n is drawn from $\mathcal{F}_{n,S_n}$ where $\gamma(n)$ is the negligible function from Lemma 5.1.5. Markov's inequality then implies that the probability (over the choice of f_n) that the event $\mathbf{E}_{n,F}$ happens is at most $1/n^2$. Let $\mathbf{E}_{\infty,F}$ denote the event that an ensemble $\{f_n\}_{n\geq 1}$ is picked such that for infinitely many n, the event $\mathbf{E}_{n,F}$ happens. Since $\sum_{n\geq 1}\frac{1}{n^2}$ converges, the Borel-Cantelli Lemma implies that $\mathbb{P}_{\{f_n\}_{n\geq 1}}[\mathbf{E}_{\infty,F}] = 0$.

Let Ω_F be the set of function ensembles for which the event $\mathbf{E}_{\infty,F}$ occurs. Ω_F has measure 0 for any F (i.e. probability zero that $\mathbf{E}_{\infty,F}$ occurs). As there are countably many PPT adversaries F, the set $\Omega = \cup_F\Omega_F$ also has measure 0. In particular, this means that for measure 0 of the function ensembles the event "$\vee_F\mathbf{E}_{\infty,F}$" happens. So setting $\gamma'(n) = n^2\gamma(n) \in \mathsf{neg}(n)$, we have that with probability 1 over the choice of function ensemble no PPT adversary can invert f_n on a non-negligible fraction of the inputs for infinitely many n. ■

5.2 The case of $t = 2$

5.2.1 The Separation Oracle Breaker

For any ensemble $\{f_n\}_{n\geq 1}, f : \{0,1\}^n \to \{0,1\}^n$ consider PRG constructions of the form $g(x_1, x_2, f(x_1), f(x_2))$. In this section for any such function ensemble $\{g_n\}_{n\geq 1}$, $g : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ we proceed as in the case of $t = 1$ to define a family of separation oracles Breaker such that using an oracle from the

family we can distinguish for infinitely many n whether $g_n(x_1, x_2, f_n(x_1), f_n(x_2))$ is pseudorandom for $x_1, x_2 \sim \mathcal{U}_n$ independently, while there exists one hard function ensemble relative to the oracle.

Again the underlying idea is the same as before. If $w \in \mathsf{Im}(g_n)$ is such that it is very likely to occur then we just output Pseudorandom. Otherwise we look at how much information w gives about x_1 (or x_2). If it is much more than $k_n + \varepsilon' n$ bits where k_n is the degeneracy of the function f_n and $\varepsilon' > 0$ is a small constant, then we know that adversary knows a significant portion of x_1 (or x_2) and so we can safely answer the query without worrying that an adversary might be able to use the answer to invert the function f. However, there is one subtlety in the above which makes it quite different from the case of $t = 1$. The adversary might not know anything about x_1 but it might be able to choose x_2 such that it gets a lot of information about x_1 from Breaker and use it to invert $f(x_1)$. Consequently, one has to ensure here that no such x_2 (or symmetrically x_1) exists.

Utilizing the above intuition we now define the family Breaker (Algorithm 4). The family is parametrized by $\omega(\log n)/n = \varepsilon < 1/1000$ and a list of integers $\{k_n\}_{n\geq 1}$ where $k_n \in [\varepsilon n, n - 2\varepsilon n]$. Furthermore, to felicitate readability in the proofs we label blocks of code of $\mathsf{Breaker}^{f,g,S}_{k,\varepsilon}(1^n, w)$ as $\mathsf{Breaker}{+}^{g,S}_{k,\varepsilon}(1^n, w)$ and $\mathsf{Breaker}{-}^{f,g}_{k,\varepsilon}(1^n, w)$. Note that the condition ★★ in $\mathsf{Breaker}{-}^{f,g}_{k,\varepsilon}(1^n, w)$ checks exactly what is described above while the condition ★ in $\mathsf{Breaker}{+}^{g,S}_{k,\varepsilon}(1^n, w)$ weeds out those $w \in \mathsf{Im}(g_n)$ which have large probability mass as we shall describe later.

Algorithm 4: $\mathsf{Breaker}^{f,g,S}_{k,\varepsilon}(1^n, w)$:

// In the following $X_1, X_2 \sim \mathcal{U}_n$, $Y_1, Y_2 \sim_{\mathcal{U}} S_n$

begin // $\mathsf{Breaker}{+}^{g,S}_{k,\varepsilon}(1^n, w)$
  for all $y_1, y_2 \in S_n$ **do**
    if $H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, y_1, Y_2) = w) \geq 2n - 2k_n + \varepsilon n$ ***or***
    $H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, Y_1, y_2) = w) \geq 2n - 2k_n + \varepsilon n$ (★) **then**
      return Pseudorandom;

begin // $\mathsf{Breaker}{-}^{f,g}_{k,\varepsilon}(1^n, w)$
  for all $y_1, y_2 \in S_n, y_1 \neq y_2$ **do**
    if $H_0(X_1 \mid \exists x_2' : g_n(X_1, x_2', y_1, f_n(x_2')) = w) \leq n - k_n - \frac{\varepsilon}{2}n$ ***and***
    $H_0(X_2 \mid \exists x_1' : g_n(x_1', X_2, f_n(x_1'), y_2) = w) \leq n - k_n - \frac{\varepsilon}{2}n$ (★★) **then**
      for all $x_1, x_2 \in \{0,1\}^n$ **do**
        if $g_n(x_1, x_2, y_1, y_2) = w$ ***and*** $f_n(x_1) = y_1$ ***and*** $f_n(x_2) = y_2$ **then**
          return (x_1, x_2);

return $\perp$

5.2.2 The Hard Distribution of Functions and a Simplified Breaker

Consider the following distribution of functions from $\{0,1\}^n$ to $\{0,1\}^n$: for an integer $k_n \in [\varepsilon n, n - 2\varepsilon n]$ which we shall choose later, pick a set $S_n \subseteq \{0,1\}^n, |S_n| = 2^{n-k_n}$. Define the distribution $\mathcal{F}_{n,S_n}$ to be the distribution induced by uniform random functions from $\{0,1\}^n$ to S_n. Note that Lemma

2.4.2 implies that with probability at least $1 - e^{-n^2}$ over the choice of f_n from the distribution $\mathcal{F}_{n,S_n}$, the number of preimages for any $y \in S_n$ is between 2^{k_n-1} and 2^{k_n+1}.

For every $n \geq 1$, we would like to pick a value of k_n such that with high probability any output of g lands in one of the cases where separation oracle Breaker returns an answer. Ideally, we would like to do this by just analyzing the structure of g but the conditions ★★ in $\mathsf{Breaker}^{f,g}_{k,\varepsilon}(1^n, w)$ involve f in a very intricate manner preventing us from doing so. To circumvent the above we first give a simpler but possibly a slightly weaker version of Breaker.

To this effect, for any $y_1 \in \{0,1\}^n$ and $w \in \{0,1\}^{2n+1}$, let $z_{y_1,w}$ denote the number of triples $(x'_1, x'_2, y'_2) \in \{0,1\}^n \times \{0,1\}^n \times S_n$ such that $g_n(x'_1, x'_2, y_1, y'_2) = w$. Let $Z^{f_n}_{y_1,w}$ be the random variable counting the pairs (x'_1, x'_2) such that $g(x'_1, x'_2, y_1, f_n(x'_2)) = w$. Note that since f_n is a uniform random function, the random variable $Z^{f_n}_{y_1,w}$ is binomially distributed with $\mathbb{E}[Z^{f_n}_{y_1,w}] = 2^{-(n-k_n)} z_{y_1,w}$. In fact, this suggests that we should be able to reduce the condition ★★ in $\mathsf{Breaker}^{-}_{k,\varepsilon}$ to checking whether $H_0(X_1X_2Y_2 \mid g(X_1, X_2, y_1, Y_2) = w) \leq 2n - 2k - \varepsilon n$ holds. Materializing the above we get the simplified oracle $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ presented in Algorithm 5. To give black-box separation, we will work with the simplified oracle $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$.

However, before proceeding we prove a simple lemma relating $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ to $\mathsf{Breaker}_{k,\varepsilon}$. This will come handy in some proofs later even though we will be using $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ for black-box separation.

Algorithm 5: $\mathsf{S\text{-}Breaker}^{f,g,S}_{k,\varepsilon}(1^n, w)$:

```
// In the following X_1, X_2 ~ U_n, Y_1, Y_2 ~_U S_n
begin // S-Breaker+^{g,S}_{k,ε}(1^n, w)
    for all y_1, y_2 ∈ S_n do
        if H_0(X_1X_2Y_2 | g_n(X_1, X_2, y_1, Y_2) = w) ≥ 2n − 2k + εn or
        H_0(X_1X_2Y_2 | g_n(X_1, X_2, Y_1, y_2) = w) ≥ 2n − 2k + εn      (S-★) then
            return Pseudorandom;

begin // S-Breaker−^{f,g}_{k,ε}(1^n, w)
    for all y_1, y_2 ∈ S_n, y_1 ≠ y_2 do
        if H_0(X_1X_2Y_2 | g_n(X_1, X_2, y_1, Y_2) = w) ≤ 2n − 2k − εn and
        H_0(X_1X_2Y_2 | g_n(X_1, X_2, Y_1, y_2) = w) ≤ 2n − 2k − εn      (S-★★) then
            for all x_1, x_2 ∈ {0,1}^n do
                if g_n(x_1, x_2, y_1, y_2) = w and f_n(x_1) = y_1 and f_n(x_2) = y_2 then
                    return (x_1, x_2);
return ⊥
```

5.2.1 Lemma.
Fix any $\omega(\log n/n) = \varepsilon < 1/1000$. Let $\{g_n\}_{n\geq 1}$, $g : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ be a function ensemble. Let $w \in \{0,1\}^{2n+1}$ satisfy the condition S-★★ in $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ for $y_1 \neq y_2 \in \{0,1\}^n$. Then with probability at least $1 - e^{-n^2}$, w also satisfies the condition ★★ for y_1, y_2 in $\mathsf{Breaker}_{k,\varepsilon}$ when f_n is drawn from the distribution $\mathcal{F}_{n,S_n}$.

▶ **Proof.** The proof follows from a standard application of Chernoff bounds and some case analysis. Let $w \in \{0,1\}^{2n+1}$ be a bit string which satisfies the condition S-★★ in S-Breaker$_{k,\varepsilon}$ for $y_1, y_2 \in \{0,1\}^n, y_1 \neq y_2$. We assume that it satisfies the first clause that is $H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, y_1, Y_2) = w) \leq 2n - 2k_n - \varepsilon n$ since the analysis for the other case is symmetric. For any $y_1 \in \{0,1\}^n$ let $z_{y_1,w}$ and $Z^{f_n}_{y_1,w}$ be as defined above. Note that w satisfying condition S-★★ in S-Breaker$_{k,\varepsilon}$ implies that $z_{y_1,w} \leq 2^{2n-2k_n-\varepsilon n}$. Next we consider the following cases:

- ($z_{y_1,w} \leq 2^{n-k_n-\varepsilon n}$). In this case $Z^{f_n}_{y_1,w} \leq 2^{n-k_n-\varepsilon n}$ with probability 1 over the choice of f_n and so it follows that $\exists x_2 \ : \ H_0(X_1 \mid g_n(X_1, X_2, y_1, f_n(X_2) = w) \leq n - k_n - \frac{\varepsilon}{2}n$.
- ($2^{n-k_n-\varepsilon n} \leq z_{y_1,w} \leq 2^{n-k_n}$). In this case $2^{-\varepsilon n} \leq \mathbb{E}[Z^{f_n}_{y_1,w}] \leq 1$ and so the probability that condition in line ★★ in Breaker$_{k,\varepsilon}$ is not satisfied is at most $\mathbb{P}\left[Z^{f_n}_{y_1,w} \geq 2^{n-k_n-\frac{\varepsilon}{2}n}\,\mathbb{E}[Z^{f_n}_{y_1,w}]\right] \leq e^{-n^3}$ by Chernoff bounds since $n - k_n \geq 2\varepsilon n$.
- ($2^{n-k_n} \leq z_{y_1,w} \leq 2^{2n-2k_n-\varepsilon n}$). It follows that $1 \leq \mathbb{E}[Z^{f_n}_{y_1,w}] \leq 2^{n-k_n-\varepsilon n}$. So the probability that the event ★★ in Breaker$_{k,\varepsilon}$ is not satisfied is at most $\mathbb{P}\left[Z^{f_n}_{y_1,w} \geq 2^{\frac{\varepsilon}{2}n}\,\mathbb{E}[Z^{f_n}_{y_1,w}]\right] \leq e^{-n^3}$.

Taking a union bound over all $y_1 \neq y_2 \in \{0,1\}^n, w \in \{0,1\}^{2n+1}$, it follows that if any $w \in \{0,1\}^{2n+1}$ satisfies S-★★ for y_1, y_2, then it also satisfies ★★ for y_1, y_2 with probability at least $1 - e^{-n^2}$ over the choice of f_n. ■

Now we come back to the question of choosing the degeneracy $k_n \in [\varepsilon n, n - 2\varepsilon n]$ of the function f_n. We believe that for any PRG candidate $\{g_n\}_{n\geq 1}$, $g : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ it should be always possible to choose an appropriate value of k_n (and corresponding set S_n) such that we can build a distinguisher (using S-Breaker$_{k,\varepsilon}$) to break the pseudorandomness of g. To this effect we posit the following conjecture. Let $y_1, y_2 \in S_n \subseteq \{0,1\}^n$, and $w \in \{0,1\}^{2n+1}$ let $\mathbf{E}^1_\varepsilon(y_1, w)$ and $\mathbf{E}^2_\varepsilon(y_2, w)$ denote the following events where X_1, X_2 are defined over the set $\{0,1\}^n$, Y_1, Y_2 over S_n and $k_n \in [\varepsilon n, n - 2\varepsilon n]$:

$$\begin{aligned}
\mathbf{E}^1_\varepsilon(y_1, w) &:= H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, y_1, Y_2) = w) \in (2n - 2k_n - \varepsilon n, 2n - 2k_n + \varepsilon n) \\
&\quad \wedge H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, Y_1,) = w) < 2n - 2k_n + \varepsilon n \\
\mathbf{E}^2_\varepsilon(y_2, w) &:= H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, y_1, Y_2) = w) < 2n - 2k_n + \varepsilon n \\
&\quad \wedge H_0(X_1X_2Y_2 \mid g(X_1, X_2, Y_1, y_2) = w) \in (2n - 2k_n - \varepsilon n, 2n - 2k_n + \varepsilon n)
\end{aligned}$$

Let $x_1, x_2 \in \{0,1\}^n$ and set $w = g_n(x_1, x_2, f_n(x_1), f_n(x_2))$. Then note that if $f_n(x_1) \neq f_n(x_2)$ holds together with the complement of the event $\mathbf{E}^1_\varepsilon(f_n(x_1), w) \vee \mathbf{E}^2_\varepsilon(f_n(x_1), w)$, then S-Breaker$_{k,\varepsilon}$ returns an answer. With this is mind the conjecture can be formulated as follows.

5.2.2 Conjecture.
There is a $\varepsilon = \varepsilon(n)$, $\omega(\log n/n) = \varepsilon < 1/1000$ which satisfies the following. For any function $g_n : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ there exists an integer $k_n \in [\varepsilon n, n - 2\varepsilon n]$ and a set S_n of size 2^{n-k_n} such that with probability at least $1 - n^{-2}$ over the choice of $f_n \sim \mathcal{F}_{n,S_n}$ the following holds: Let $x_1, x_2 \sim \mathcal{U}_n$ and $w := g_n(x_1, x_2, f_n(x_1), f_n(x_2))$. Then

$$\mathbb{P}_{x_1,x_2}\left[\neg\left(\mathbf{E}^1_\varepsilon(f_n(x_1), w) \vee \mathbf{E}^2_\varepsilon(f_n(x_1), w)\right)\right] \geq 3/4. \tag{5.5}$$

□

We remark that the value $3/4$ in the statement above is arbitrary. In fact, we can replace it by $1/2 + \beta(n)$ for any non-negligible function $\beta(n)$ and the following proofs still hold.

Since f_n is a uniform random function from $\{0,1\}^n$ to S_n, the statement above essentially states that for any PRG candidate g there always exist $\omega(\log n/n) = \varepsilon < 1/1000$, $k_n \in [\varepsilon n, n - 2\varepsilon n]$ and a set $S_n \subseteq \{0,1\}^n, |S| = 2^{n-k_n}$ such that if we take x_1, x_2 drawn u.a.r. from $\{0,1\}^n$ and y_1, y_2 u.a.r from S_n, then with high probability one of conditions S-★ or S-★★ are satisfied for $g_n(x_1, x_2, y_1, y_2)$. Observe that this is some sort of generalization of Equation (5.1) from the case for $t = 1$. The probability over f_n in the statement above is a technical annoyance since we need it to apply the Borel-Cantelli Lemma. However, we hope that it should be possible to circumvent this.

For the following parts we shall assume that for all n large enough , Conjecture 5.2.2 gives us a suitable ε, k_n and a subset S_n of size 2^{n-k_n} which satisfy the statement of the conjecture. Let $k = \{k_n\}_{n\geq 1}$ and $S = \{S_n\}_{n\geq 1}$. We shall use $\mathsf{S\text{-}Breaker}^{f,g,S}_{k,\varepsilon}$ as our separation oracle.

5.2.3 The Distinguisher

Consider the following simple distinguisher $G^{\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ for distinguishing $g_n(X_1, X_2, f_n(X_1), f_n(X_2))$ for $X_1, X_2 \sim \mathcal{U}_n$ from $\mathcal{U}_{2n+1}$: Output 1 if $\mathsf{S\text{-}Breaker}^{f,g,S}_{k,\varepsilon}(1^n, w)$ does not return $\perp$. Assuming Conjecture 5.2.2 we now show that this distinguisher has a non-negligible distinguishing probability.

5.2.3 Lemma.
Let $\{f_n\}_{n\geq 1}, f_n : \{0,1\}^n \to \{0,1\}^n$ be a function ensemble where f_n is drawn from the distribution $\mathcal{F}_{n,S_n}$. Then there exists a non-negligible function $\delta(n)$ such that with probability at least $1 - n^{-2}$ over the choice of f, the distinguishing advantage for the above distinguisher $G^{\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ on security parameter 1^n is at least $\delta(n)$.

▶ **Proof.** The proof is similar to that of Lemma 5.1.3. Let us first condition on the event that f_n satisfies Conjecture 5.2.2 and also that number of pre-images for each $y \in \mathsf{Im}(f_n)$ is between 2^{k_n-1} and 2^{k_n+1}. Note that f_n satisfies both the above the above properties with probability at least $1 - n^{-2} - e^{-n^2}$ due to Conjecture 5.2.2 and Lemma 2.4.2.

Let $x_1, x_2 \sim \mathcal{U}_n$ and $w := g_n(x_1, x_2, f_n(x_1), f_n(x_2))$. Note that Conjecture 5.2.2 implies that the following holds :

$$\mathbb{P}_{x_1,x_2}\left[\neg\left(\mathbf{E}^1_\varepsilon(f_n(x_1), w) \vee \mathbf{E}^2_\varepsilon(f_n(x_1), w)\right)\right] \geq 3/4.$$

Also note that the distinguisher surely outputs 1 if the above condition $\neg\left(\mathbf{E}^1_\varepsilon(f_n(x_1), w) \vee \mathbf{E}^2_\varepsilon(f_n(x_1), w)\right)$ holds together with $f_n(x_1) \neq f_n(x_2)$. The probability that $f_n(x_1) \neq f_n(x_2)$ for $x_1, x_2 \sim \mathcal{U}_n$ is at most $2^{-(n-k_n)+1} \leq 2^{-\varepsilon n/2}$ since f is approximately k-regular. It follows that the distinguisher outputs 1 on $w := g_n(x_1, x_2, f_n(x_1), f_n(x_2))$ for $x_1, x_2 \sim \mathcal{U}_n$ with probability at least $5/8$ for large enough n.

On the other hand if $w \sim \mathcal{U}_{2n+1}$, then we claim that the distinguisher will output 1 with probability at most $\frac{1}{2} + 2^{-\varepsilon n}$. To see this first note that at most $\frac{1}{2}$ fraction of the $w \in \{0,1\}^{2n+1}$ are such that $\mathsf{S\text{-}Breaker}_{-k,\varepsilon}$ is invoked and it returns an answer. This is so because $\mathsf{S\text{-}Breaker}_{-k,\varepsilon}$ checks whether w is in the set $\{g_n(x_1, x_2, f_n(x_1), f_n(x_2)) \mid x_1, x_2 \in \{0,1\}^n\}$ and this set has size at most 2^{2n}.

For the cases where $\mathsf{S\text{-}Breaker}_{+k,\varepsilon}$ is invoked and returns Pseudorandom as the answer, define the set $W \subseteq \mathsf{Im}(g_n)$ to be the set of $w \in \mathsf{Im}(g)$ which satisfy the condition S-★ for some y_1 or y_2. We

now claim that $|W| \leq 2^{2n-\varepsilon n}$ from which it follows that the probability that S-Breaker$+_{k,\varepsilon}$ outputs Pseudorandom is at most $|W|/2^{2n+1} \leq 2^{-\varepsilon n}$.

To see the above claim, note that for any $y \in S_n$ the number of $w \in W$ which satisfies either $H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, y, Y_2) = w) \geq 2n - 2k_n + \varepsilon n$ or $H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, Y_1, y) = w) \geq 2n - 2k_n + \varepsilon n$ is at most

$$\frac{|\mathrm{Supp}(X_1X_2Y_1)|}{2^{2n-2k_n+\varepsilon n}} = \frac{2^{3n-k_n}}{2^{2n-2k_n+\varepsilon n}} \leq 2^{n+k_n-\varepsilon n}.$$

Since $|S_n| = 2^{n-k_n}$, we have that $|W| \leq 2^{n+k_n-\varepsilon n}|S_n| \leq 2^{2n-\varepsilon n}$. This completes the proof. ■

Applying the Borel-Cantelli Lemma to the above lemma we get the following.

5.2.4 Lemma.
For any function ensemble $\{g_n\}_{n\geq 1}, g_n : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ there exists a non-negligible function $\delta'(n)$ and a distinguisher $G^{\mathsf{Breaker}_{k,\varepsilon}}$ such that with probability 1 over the choice of function ensembles $\{f_n\}_{n\geq 1}$ where f_n is drawn as per the distribution $\mathcal{F}_{n,S_n}$, the following holds

$$\left|\mathbb{P}_{x_1,x_2\sim\mathcal{U}_n}[G^{\text{S-Breaker}_{k,\varepsilon}}(1^n, g_n(x_1, x_2, f_n(x_1), f_n(x_2))) = 1] - \mathbb{P}_{w\in\mathcal{U}_{2n+1}}[G^{\text{S-Breaker}_{k,\varepsilon}}(1^n, w) = 1]\right| \geq \delta'(n).$$

5.2.4 f remains secure relative to Breaker

We now show that a function drawn according to the distribution $\mathcal{F}_{n,S_n}$ defined above remains hard with respect to S-Breaker.

5.2.5 Lemma.
Let $\{f_n\}_{n\geq 1}, f_n : \{0,1\}^n \to \{0,1\}^n$ be a function ensemble where f_n is drawn according to the distribution $\mathcal{F}_{n,S_n}$ defined above and let $F^{f,\text{S-Breaker}_{k,\varepsilon}}$ be any probabilistic polynomial time adversary that takes as input $(1^n, y)$ where $y \in \{0,1\}^n$ and has oracle access to S-Breaker$_{k,\varepsilon}$ and f. Then there exists a negligible function $\gamma(n)$ such that the following holds.

$$\mathbb{P}_{f\sim\mathcal{F}_n, y\sim\mathcal{U}_n}[F^{f,\text{S-Breaker}_{k,\varepsilon}}(1^n, y) \in f_n^{-1}(y)] \leq \gamma(n).$$

The proof is similar to the proof of Lemma 5.1.5. Before embarking upon it, lets first state the following lemma which follows by applying the Borel-Cantelli Lemma to the above.

5.2.6 Lemma.
There exists a negligible function $\gamma'(n)$ such that with probability 1 over the choice of function ensembles $\{f_n\}_{n\geq 1}$ where f_n is drawn u.a.r. from the distribution $\mathcal{F}_{n,S_n}$ the following holds for all probabilistic polynomial time adversaries $F^{f,\text{Breaker}_{k,\varepsilon}}$ (with oracle access to S-Breaker$_{k,\varepsilon}$ and f) and large enough n.

$$\mathbb{P}_{y\sim\mathcal{U}_n}[F^{f,\text{S-Breaker}_{k,\varepsilon}}(1^n, y) \in f_n^{-1}(y)] \leq \gamma'(n).$$

Assuming Conjecture 5.2.2, the above lemma together with Lemma 5.2.3 proves the following Theorem which is the main result of this section.

▶ **Theorem. 5.2.7 (Impossibility of Black-box PRG construction with 2 queries)**
Assuming Conjecture 5.2.2, there does not exist any oracle function ensemble $\{g_n\}_{n\geq 1}, g^{(\cdot)} : \{0,1\}^{2n} \to \{0,1\}^{2n+1}$ *such that the following holds for every function ensemble* $\{f_n\}_{n\geq 1}, f : \{0,1\}^n \to \{0,1\}^n$ *for infinitely many n.*

- *For any* $x_1, x_2 \in \{0,1\}^n$, $g_n^{f_n}(x_1, x_2) = g_n(x_1, x_2, f_n(x_1), f_n(x_2))$.
- *If f is secure, then* $g^f(\mathcal{U}_n, \mathcal{U}_n)$ *is computationally indistinguishable from* $\mathcal{U}_{2n+1}$ *with a black-box proof of security.*

Now we proceed with the proof of Lemma 5.2.5. Let $F^{f,\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ be a polynomial time adversary with oracle access to $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ and f. The adversary gets as input a security parameter 1^n and $y \in \{0,1\}^n$ and tries to find $f_n^{-1}(y)$. Consider the change in the execution of the adversary if we perturb f by randomly choosing an $x^* \in \{0,1\}^n$ and mapping it to y. Lets denote this new function by $f_{*,n}$ and the corresponding ensemble generated from f by replacing f_n by $f_{*,n}$. As before we show that only with a very small probability there will be a change in the execution of the adversary when it uses the oracles $\mathsf{S\text{-}Breaker}^{f_*}$ and f_* instead of $\mathsf{Breaker}^f$ and f.

5.2.8 Lemma. ($\mathcal{F}_{n,S_n}$ remains hard)
The probability over the choice of x^* *and f that the execution of* $F^{f,\mathsf{S\text{-}Breaker}^f}(1^n, y)$ *differs from the execution of* $F^{f_*,\mathsf{S\text{-}Breaker}^{f_*}}(y)$ *is at most* $\mathsf{poly}(n)2^{-k_n - \varepsilon n/2} + e^{-n^2}$.

Note that the above lemma implies Lemma 5.2.5. The proof for this is almost verbatim to the proof of Lemma 5.1.5 from Lemma 5.1.6. We now prove the above lemma. Without loss of generality we can assume that F is deterministic (similar to the proof of Lemma 5.1.5).

▶ **Proof. (Lemma 5.2.8)** We assume that $x^* \notin f^{-1}(y)$, since the statement holds trivially otherwise.

Consider the execution of $F^{f,\mathsf{S\text{-}Breaker}^f_{k,\varepsilon}}(1^n, y)$ and $F^{f_*,\mathsf{S\text{-}Breaker}^{f_*}_{k,\varepsilon}}(1^n, y)$ in parallel. $F^{f,\mathsf{S\text{-}Breaker}^f_{k,\varepsilon}}$ makes queries to f and $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ and computes an answer, while $F^{f_*,\mathsf{S\text{-}Breaker}^{f_*}_{k,\varepsilon}}$ queries f_* and $\mathsf{S\text{-}Breaker}^{f_*}_{k,\varepsilon}$. Since F is deterministic, the only way the execution can differ is when a particular query to f (or $\mathsf{Breaker}^f_{k,\varepsilon}$) returns a different answer than the same query to f_* (or $\mathsf{S\text{-}Breaker}^{f_*}_{k,\varepsilon}$).

Let us now bound the probability that this happens for a fixed query assuming all the previous queries returned the same answers in both executions.

First note that, all queries made to f (or f_*) or $\mathsf{S\text{-}Breaker}^f_{k,\varepsilon}$ (or $\mathsf{S\text{-}Breaker}^{f_*}_{k,\varepsilon}$) for a security parameter other than 1^n return the same answer in both cases. So let us turn our attention to queries made only with security parameter 1^n. In this case, the answers to one of the queries to f versus f_* are different only when the adversary queries with x^*. The probability that this happens for a particular query is at most 2^{-n}.

Next we upper bound the probability that any fixed $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ query is the first one where a difference in the answers returned by $\mathsf{S\text{-}Breaker}^f_{k,\varepsilon}$ and $\mathsf{S\text{-}Breaker}^{f_*}_{k,\varepsilon}$ occurs. First note that the execution of $\mathsf{S\text{-}Breaker}+_{k,\varepsilon}$ is not affected by the change so if the $\mathsf{S\text{-}Breaker}^f_{k,\varepsilon}$ returns Pseudorandom,

then a query to S-Breaker$^{f^*}_{k,\varepsilon}$ will also return the same. So, the answers to S-Breaker$_{k,\varepsilon}$ queries can be different in the two executions above only if one of the following happens :

- The adversary queries with input $w \in \{0,1\}^{2n+1}$ and S-Breaker$^{f}_{k,\varepsilon}$ either returned $\perp$ or (x, x_2) where $x \in f_n^{-1}(y)$ and $x_2 \in \{0,1\}^n$ (or symmetrically (x_1, x) for some $x \in f_n^{-1}(y), x_1 \in \{0,1\}^n$), while the query to S-Breaker$^{f^*}_{k,\varepsilon}$ returned (x^*, x'_2) (or (x'_1, x^*)) for some $x'_1, x'_2 \in \{0,1\}^n$. Note that this happens exactly if the following holds:

$$H_0(X_1X_2Y_2 \mid g_n(X_1, X_2, y, Y_2) = w) \leq 2n - 2k_n - \varepsilon n, \text{ and}$$
$$H_0(X_1X_2Y_1 \mid g_n(X_1, X_2, Y_1, y) = w) \leq 2n - 2k_n - \varepsilon n,$$

  and $g_n(x^*, x'_2, y, f_n(x'_2)) = w$ (or $g_n(x'_1, x^*, f_n(x'_1), y) = w$). Lemma 5.2.1 implies that in this case with probability at least $1 - e^{-n^2}$ over the choice of f_n, both $H_0(X_1 \mid \exists x'_2\ g_n(X_1, x'_2, y, f_n(x'_2)) = w) \leq n - k_n - \frac{\varepsilon}{2}n$ for $X_1 \sim \mathcal{U}_n$ and $H_0(X_2 \mid \exists x'_1\ g_n(x'_1, X_2, f_n(x'_1), y_2) = w) \leq n - k - \frac{\varepsilon}{2}n$ for $X_2 \sim \mathcal{U}_n$ hold. Since x^* is chosen independently and uniformly at random, the above implies that the probability that this case occurs is at most $\frac{1}{2^n}|\{x_1 \in \{0,1\}^n \mid \exists x'_2 : g(x_1, x'_2, y, f(x'_2)) = w\}| \leq 2^{-k_n - \varepsilon n/2}$ (the other case where (x'_1, x^*) is returned is symmetric.)

- The adversary queries with some $w \in \{0,1\}^{2n+1}$, S-Breaker$^{f}_{k,\varepsilon}$ outputs (x^*, x_2) for some $x_2 \in \{0,1\}^n$ (or (x_1, x^*) for some $x_1 \in \{0,1\}^n$) while S-Breaker$^{f^*}_{k,\varepsilon}$ outputs something else. Let $(x'_1, x'_2) := \text{S-Breaker}^{f}_{k,\varepsilon}(1^n, w)$ where $x'_1, x'_2 \in \{0,1\}^n$ (note that in this case S-Breaker$^{f}_{k,\varepsilon}$ does not return $\perp$). Then in this case we have that $x^* = x'_1$ (or $x^* = x'_2$) and hence the probability that this event happens is at most 2^{-n}.

The proof is then completed by considering all the queries to f and S-Breaker$_{k,\varepsilon}$ simultaneously and using the union bound to bound the probability that the one of them is the first one where the execution differs. ■

5.3 Generalization for $t > 2$

In this section we generalize the approach from the last section for the cases where $2 < t \leq n$. We show that a suitable generalization of Conjecture 5.2.2 is sufficient to show black-box impossibility for these constructions also. We furthermore believe that it should be possible to show that the generalized conjecture holds for t up to $n/\log^2 n$. Note that this limit on t is arbitrary (but motivated by some underlying intuition) and it could very well be that the conjecture is true for even higher values of t.

Most of the proofs in this section are straightforward generalization of proofs from last section. We will omit most of them and point out only the changes that need to be made in most cases. We start by describing the separation oracle.

5.3.1 The Separation Oracle Breaker

Let us first fix some notational shorthands. We will use $x_{[t]} = (x_1, \ldots, x_t)$ where each $x_i \in \{0,1\}^n$ to describe a t-tuple of strings of length n. Similarly, we use $X_{[t]}$ to denote a tuple of t-random variables.

Moreover, we write $X_{[t]\setminus\{i\}}$ to denote the $(t-1)$-tuple which we get after dropping X_i from the tuple. Also we use $f(x_{[t]})$ to denote $(f(x_1),\ldots,f(x_t))$.

For any ensemble $\{f_n\}_{n\geq 1}, f_n : \{0,1\}^n \to \{0,1\}^n$, here we consider PRG candidates of the form $g(x_1,\ldots,x_t,f(x_1),\ldots,f(x_t))$. Given any such ensemble $\{g_n\}_{n\geq 1}, g_n : \{0,1\}^{tn} \to \{0,1\}^{tn+1}$, we define a family of separation oracles Breaker in Algorithm 6. Again the family is parametrized by a $\omega(\log n/n) < \varepsilon < 1/1000$ and a list of integers $\{k_n\}_{n\geq 1}$ where $k_n \in [\varepsilon n, n - 2\varepsilon n]$ and blocks of Breaker are labeled as Breaker$+^{g,S}_{k,\varepsilon}(1^n, w)$ and Breaker$-^{f,g}_{k,\varepsilon}(1^n, w)$. Note that it is a straightforward generalization of Algorithm 4. The main difference in this case is that here we need to ensure whether for any $i \in [t]$ there exists any possible choices $\tilde{x} := x_1,\ldots,x_{i-1},x_{i+1},\ldots x_t$ such that an adversary could produce a suitable $w \in \{0,1\}^{tn+1}$ to query Breaker just building on information given by w, y_i and $\tilde{x}$ and get $f_n^{-1}(y_i)$ as an answer. More formally, one needs to check that for all $i \in [t]$, $H_0(X_i \mid \exists x_{[i-1]}, x_{[t-i]} : g(x_{[i-1]}, X_i, x_{[t-i]}, f(x_{[i-1]}]), y_i, f(x_{[t-i]})) = w) \leq n - k_n - \varepsilon n$ before using the Breaker to invert g.

Algorithm 6: Breaker$^{f,g,S}_{k,\varepsilon}(1^n, w)$:

// In the following $X_1,\ldots,X_t \sim \mathcal{U}_n$ and $Y_1,\ldots,Y_t \sim_{\mathcal{U}} S_n$;

begin // Breaker$+^{g,S}_{k,\varepsilon}(1^n, w)$
  for all $y_1,\ldots,y_t \in S_n$ **do**
    if $\exists i \in [t] : H_0(X_{[t]}Y_{[t]\setminus\{i\}} \mid g_n(X_{[t]}, Y_1,\ldots,Y_{i-1},y_i,Y_{i+1},\ldots,Y_t) = w) \geq t(n-k_n) + \varepsilon n$ **then**
      return Pseudorandom;

begin // Breaker$-^{f,g}_{k,\varepsilon}(1^n, w)$
  for all $y_1,\ldots,y_t \in S_n, y_i \neq y_j$ *for* $i \neq j$ **do**
    if $\forall i \in [t]\ H_0(X_i \mid \exists x_{[i-1]}, x_{[t-i]} : g_n(x_{[i-1]}, X_i, x_{[t-i]}, f_n(x_{[i-1]}]), y_i, f_n(x_{[t-i]})) = w) \leq n - k_n - \frac{\varepsilon}{2}n$ **then**
      for all $x_{[t]} \in (\{0,1\}^n)^t$ **do**
        if $g_n(x_{[t]}, y_{[t]}) = w$ ***and*** $f_n(x_{[t]}) = y_{[t]}$ **then**
          return $x_{[t]}$;

return $\perp$

5.3.2 The Hard Distribution of Functions and Simplification of Breaker

As before, the hard distribution of functions we choose is the distribution induced by uniform random functions from $\{0,1\}^n$ to a set $S_n \subseteq \{0,1\}^n, |S_n| = 2^{n-k_n}$ for a suitable $k_n \in [\varepsilon n, n - 2\varepsilon n]$ which we pick such that with high probability an output of g lands in one of the cases where separation oracle Breaker returns an answer.

Since f_n is a random function, we can simplify Breaker$_{k,\varepsilon}$ to S-Breaker$_{k,\varepsilon}$ (Algorithm 7) similar to the case for $t = 2$. We also have the following lemma relating S-Breaker$_{k,\varepsilon}$ to Breaker$_{k,\varepsilon}$.

Algorithm 7: S-Breaker$^{f,g,S}_{k,\varepsilon}(1^n, w)$:

// In the following $X_1, \ldots, X_t \sim \mathcal{U}_n$ and $Y_1, \ldots, Y_t \sim S_n$;

begin // S-Breaker$+^{g,S}_{k,\varepsilon}(1^n, w)$
  for all $y_1, \ldots, y_t \in S_n$ **do**
    if $\exists i \in [t] : H_0(X_{[t]}Y_{[t]\setminus\{i\}} \mid g_n(X_{[t]}, Y_1, \ldots, Y_{i-1}, y_i, Y_{i+1}, \ldots, Y_t) = w) \geq t(n-k_n) + \varepsilon n$ **then**
      return Pseudorandom;

begin // S-Breaker$-^{f,g}_{k,\varepsilon}(1^n, w)$
  for all $y_1, \ldots, y_t \in S_n, y_i \neq y_j$ *for* $i \neq j$ **do**
    if $\forall i \in [t] \ : \ H_0(X_{[t]}Y_{[t]\setminus\{i\}} \mid g_n(X_{[t]}, Y_1, \ldots, Y_{i-1}, y_i, Y_{i+1}, \ldots, Y_t) = w) \leq t(n-k_n) - \varepsilon n$ **then**
      for all $x_{[t]} \in (\{0,1\}^n)^t$ **do**
        if $g_n(x_{[t]}, y_{[t]}) = w$ ***and*** $f_n(x_{[t]}) = y_{[t]}$ **then**
          return $x_{[t]}$;

return $\perp$

5.3.1 Lemma.
Fix any ε satisfying $\omega(\log n/n) = \varepsilon < 1/1000$. Let $\{g_n\}_{n\geq 1}, g : \{0,1\}^m \to \{0,1\}^{m+1}$ be a function ensemble. Let $w \in \{0,1\}^{m+1}$ satisfy the condition S-★★ *in* S-Breaker$_{k,\varepsilon}$ *for $y_{[t]} \in (\{0,1\}^n)^t$ where $y_i \neq y_j, i \neq j \in [t]$. Then with probability at least $1 - e^{-n^2}$, w also satisfies condition ★★ for $y_{[t]}$ in* Breaker$_{k,\varepsilon}$ *when f_n is drawn from the distribution $\mathcal{F}_{n,S_n}$.*

▶ **Proof. (Sketch)** The proof is similar to the proof of Lemma 5.2.1 although there are some subtleties. Here we will only describe the details which are different.

For any i, fix y_i and let $z_{y_i,w}$ denote the number of tuples $(x'_1, \ldots, x'_t, y'_1, \ldots, y'_{i-1}, y_i, y'_{i+1}, \ldots, y'_t)$ where $x'_j \in \{0,1\}^n$ for all $j \in [t]$ and $y'_j \in S_n$ satisfying the following :

1. there are no collision among x'_j, *i.e.*, $x'_j \neq x'_l$ for $j \neq l$.
2. $g_n(x'_1, \ldots, x'_t, y'_1, \ldots, y'_{i-1}, y_i, y'_{i+1}, \ldots, y_t) = w$.

Also let $Z^{f_n}_{y_i}(w)$ denote the random variable counting the number of tuples $(x'_1, \ldots, x'_t, f(x'_1), f(x'_{i-1}), y_i, f(x'_{i+1}), \ldots, f(x'_t))$ mapping to w under g where again we just count tuples where there are no collisions among x'_j. Observe that in S-Breaker$_{k,\varepsilon}$ the condition S-★★ is checked only for those $y_1, \ldots y_t$ where $y_i \neq y_j$ for $i \neq j$. This implies that $Z^{f_n}_{y_i}(w)$ is distributed binomially with $\mathbb{E}_{f_n}[Z^{f_n}_{y_i}(w)] = 2^{-(t-1)(n-k_n)} z_{y_i,w}$ since each x'_j independently maps to y'_j with probability $2^{-(n-k_n)}$ in this case.

The condition S-★★ in S-Breaker$_{k,\varepsilon}$ implies that $z_{y_1,w} \leq 2^{t(n-k_n)-\varepsilon n}$ (In fact, it also counts tuples where collision might occur) and so an application of Chernoff bounds (as in Lemma 5.2.1) together with union bound proves the lemma. ■

Now we come back to the question of choosing the degeneracy k_n of the function f_n. For this we need a general version of Conjecture 5.2.2. Let $x_1, \ldots, x_t \in \{0,1\}^n$, and $w \in \{0,1\}^{tn+1}$. For better readability let us use the shorthand $H_0^i(y_i, w)$ to denote $H_0(X_{[t]}Y_{[t]\setminus\{i\}} \mid g(X_{[t]}, Y_1, \ldots, Y_{i-1}, y_i, Y_{i+1}, \ldots, Y_t) = w)$ where X_i are defined over the set $\{0,1\}^n$ and Y_i over $S_n \subseteq \{0,1\}^n$. Define $\mathbf{E}_\varepsilon^i(y_i, w)$ to be the following event for some $k_n \in [\varepsilon n, n - 2\varepsilon n]$ for a parameter $\omega(\log n/n) = \varepsilon < 1/1000$:

$$\mathbf{E}_\varepsilon^i(y_i, w) := H_0^i(y_i, w) \in \Big(t(n-k_n) + -\varepsilon n,\ t(n-k_n) + \varepsilon n\Big] \bigwedge_{j \in [t]\setminus\{i\}} H_0^j(y_j, w) < t(n-k_n) + \varepsilon n$$

With the above notation, the Conjecture 5.2.2 can be generalized as follows.

5.3.2 Conjecture.
For every $t \in \mathbb{N}$ satisfying $2 \le t \le n/\log^2 n$, there is an ε, $\omega(\log n/n) = \varepsilon < 1/1000$ which satisfies the following. For any function $g_n : \{0,1\}^{tn} \to \{0,1\}^{tn+1}$ there exists an integer $k_n \in [\varepsilon n, n - 2\varepsilon n]$, a set S_n of size 2^{n-k_n} such that with probability at least $1 - n^{-2}$ over the choice of $f_n \sim \mathcal{F}_{n,S_n}$ the following holds: Let $x_1, \ldots, x_t \sim \mathcal{U}_n$ and $w := g_n(x_1, \ldots, x_t, f_n(x_1), \ldots, f_n(x_t))$. Then

$$\mathbb{P}_{x_1,\ldots,x_t}\left[\neg\left(\bigvee_{i\in[t]} \mathbf{E}_\varepsilon^i(f_n(x_i), w)\right)\right] \ge 3/4. \tag{5.6}$$

□

5.3.3 The Distinguisher

Consider the following simple distinguisher $G^{\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ for distinguishing $g_n(X_1, X_2, \ldots, X_t, f_n(X_1), \ldots, f_n(X_t))$ for $X_1, \ldots, X_t \sim \mathcal{U}_n$ from $\mathcal{U}_{tn+1}$ for infinitely many n: Output 1 if $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ does not return $\perp$. Assuming Conjecture 5.3.2, we can show that this distinguisher has a non-negligible distinguishing probability.

5.3.3 Lemma.
Let $2 \le t \le n/\log^2 n$. Assume Conjecture 5.3.2 is true and ε, k_n and S_n be the parameters which satisfy the conjecture for t. Let $\{f_n\}_{n\ge1}, f : \{0,1\}^n \to \{0,1\}^n$ be a function ensemble where f_n is drawn according to the distribution $\mathcal{F}_{n,S_n}$. Then there exists a non-negligible function $\delta(n)$ such that with probability at least $1 - n^{-2}$ over the choice of f_n, the distinguishing advantage for the above distinguisher $G^{\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ is at least $\delta(n)$.

Applying the Borel-Cantelli Lemma to the above lemma we get the following.

5.3.4 Lemma.
Let $2 \le t \le n\log^2 n$. Assume Conjecture 5.3.2 is true and ε, k_n and S_n be the parameters which satisfy the conjecture for t. Then for any function ensemble $\{g_n\}_{n\ge1}, g_n : \{0,1\}^{tn} \to \{0,1\}^{tn+1}$ there exists a non-negligible function $\delta'(n)$ and a distinguisher $G^{\mathsf{Breaker}_{k,\varepsilon}}$ such that with probability 1 over the choice of function ensembles $\{f_n\}_{n\ge1}$ where f_n is drawn as per the distribution $\mathcal{F}_{n,S_n}$, the following holds

$$\left|\mathbb{P}_{x_1,\ldots,x_t\sim\mathcal{U}_n}[G^{\mathsf{S\text{-}Breaker}_{k,\varepsilon}}(1^n, g_n(x_1, \ldots, x_t, f_n(x_1), \ldots, f_n(x_t))) = 1] - \mathbb{P}_{w\sim\mathcal{U}_{tn+1}}[G^{\mathsf{S\text{-}Breaker}_{k,\varepsilon}}(1^n, w) = 1]\right| \ge \delta'(n).$$

5.3.4 f remains secure relative to Breaker

Following the techniques from last two sections we can show that a function drawn from the distribution $\mathcal{F}_{n,S_n}$ defined above remains hard with respect to S-Breaker.

5.3.5 Lemma.
Let $\{f_n\}_{n\geq 1}, f_n : \{0,1\}^n \to \{0,1\}^n$ be a function ensemble where f_n is drawn from the distribution $\mathcal{F}_{n,S_n}$ defined above and let $F^{f,\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ be any probabilistic polynomial time adversary that takes as input $(1^n, y)$ where $y \in \{0,1\}^n$ and has oracle access to $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ and f. Then there exists a negligible function $\varepsilon'(n)$ such that the following holds.

$$\mathbb{P}_{f_n\sim\mathcal{F}_{n,S_n},y\sim\mathcal{U}_n}[F^{f,\mathsf{S\text{-}Breaker}_{k,\varepsilon}}(1^n,y) \in f_n^{-1}(y)] \leq \varepsilon'(n).$$

The proof is similar to the proof of Lemma 5.2.5. Before sketching the proof, lets first state the following lemma which follows by applying the Borel-Cantelli Lemma to the above.

5.3.6 Lemma.
There exists a negligible function $\gamma(n)$ such that with probability 1 over the choice of function ensembles $\{f_n\}_{n\geq 1}$ where f_n is drawn u.a.r. from the family $\mathcal{F}_{n,S_n}$ such that the following holds for all probabilistic polynomial time adversaries $F^{f,\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ (with oracle access to $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ and f) and large enough n.

$$\mathbb{P}_{y\sim\mathcal{U}_n,F}[F^{f,\mathsf{S\text{-}Breaker}_{k,\varepsilon}}(1^n,y) \in f_n^{-1}(y)] \leq \gamma(n).$$

Assuming Conjecture 5.3.2, the above lemma together with Lemma 5.3.4 proves the following Theorem which is the main result of this section.

▶ **Theorem. 5.3.7 (Impossibility of Black-box PRG construction with t queries)**
Let $2 \leq t \leq n/\log^2 n$. If Conjecture 5.3.2 holds, then there does not exist any oracle function ensemble $\{g_n\}_{n\geq 1}, g^{(\cdot)} : \{0,1\}^{tn} \to \{0,1\}^{tn+1}$ such that the following holds for every function ensemble $\{f_n\}_{n\geq 1}, f : \{0,1\}^n \to \{0,1\}^n$ for infinitely many n.

- *For any $x_1, \ldots, x_t \in \{0,1\}^n$, $g_n^{f_n}(x_1, \ldots, x_t) = g_n(x_1, \ldots, x_t, f_n(x_1), \ldots, f_n(x_t))$.*
- *If f is secure, then $g^f((\mathcal{U}_n)^t)$ is computationally indistinguishable from $\mathcal{U}_{tn+1}$ with a black-box proof of security.*

Lemma 5.3.5 is proved in the same manner as Lemma 5.2.5. Let $F^{f,\mathsf{S\text{-}Breaker}_{k,\varepsilon}}$ be a probabilistic polynomial time adversary with oracle access to $\mathsf{S\text{-}Breaker}_{k,\varepsilon}$ and f. The adversary on security parameter 1^n gets as input $y \in \{0,1\}^n$ and tries to find $f_n^{-1}(y)$. Consider the change in the execution of the adversary if we perturb f_n by randomly choosing an $x^* \in \{0,1\}^n$ and mapping it to y. Lets denote this new function by $f_{*,n}$ and the corresponding ensemble generated from f by replacing f_n with $f_{*,n}$ by f_*. One can generalize the proof of Lemma 5.2.5 and show that only with a very small probability there will be a change in the execution of the adversary when it uses the oracles $\mathsf{Breaker}_*^f$ and f_* instead of Breaker and f.

5.3.8 Lemma. ($\mathscr{F}_{n,S_n}$ remains hard)
The probability over the choice of x^* and f_n that the execution of $F^{f,\mathsf{S\text{-}Breaker}^f_{k,\varepsilon}}(1^n, y)$ differs from the execution of $F^{f_*,\mathsf{S\text{-}Breaker}_{k,\varepsilon}{}^{f_*}}(1^n, y)$ is at most $\mathsf{poly}(n)2^{-k_n-\varepsilon n} + e^{-n^2}$.

Note that the above lemma implies Lemma 5.3.5. The proof is almost verbatim to the proof of Lemma 5.1.5 from Lemma 5.1.6.

Bibliography

[1] S. Arora and B. Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. ISBN 978-0-521-42426-4.

[2] B. Barak, M. Hardt, and S. Kale. The uniform hardcore lemma via approximate bregman projections. In *SODA*, pages 1193–1200, 2009.

[3] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984. ISSN 0097-5397.

[4] J. L. Carter and M. N. Wegman. Universal classes of hash functions (extended abstract). In *STOC '77: Proceedings of the ninth annual ACM symposium on Theory of computing*, pages 106–112, New York, NY, USA, 1977. ACM.

[5] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *FOCS*, pages 305–313, 2000.

[6] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0521791723.

[7] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, New York, NY, USA, 1989. ACM. ISBN 0-89791-307-8.

[8] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33: 792–807, August 1986. ISSN 0004-5411.

[9] O. Goldreich, H. Krawczyk, and M. Luby. On the existence of pseudorandom generators. *SIAM J. Comput.*, 22(6):1163–1175, 1993.

[10] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:12–24, 1999.

[11] I. Haitner. *New Implications and Improved Efficiency of Constructions Based on One-way Functions*. PhD thesis, Weizmann Institute of Science, 2008.

[12] I. Haitner and T. Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, pages 202–219, 2009.

[13] I. Haitner, D. Harnik, and O. Reingold. On the power of the randomized iterate. In *In 21st ACM Symposium on the Theory of Computing*, pages 22–40. Springer, 2006.

[14] I. Haitner, O. Reingold, and S. P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *STOC*, pages 437–446, 2010.

[15] J. Håstad. Pseudo-random generators under uniform assumptions. In *STOC*, pages 395–404, 1990.

[16] T. Holenstein. Key agreement from weak bit agreement. In *STOC*, pages 664–673, 2005.

[17] T. Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *TCC*, pages 443–461, 2006.

[18] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstracts). In *STOC*, pages 12–24, 1989.

[19] L. A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987. ISSN 0209-9683. doi: http://dx.doi.org/10.1007/BF02579323.

[20] M. Naor. Bit commitment using pseudo-randomness. *Journal of Cryptology*, 4:151–158, 1991.

[21] A. A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:204–213, 1994.

[22] O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In M. Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.

[23] D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions. In *Advances in Cryptology — EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer-Verlag, 1998.

[24] A. C. Yao. Theory and application of trapdoor functions. In *SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 80–91, Washington, DC, USA, 1982. IEEE Computer Society.