HARVARD UNIVERSITY

On Resilient and Exposure-Resilient Functions

by

Yakir Reshef

A thesis
presented in partial fulfillment
of the requirements for the degree
of Bachelor of Arts with Honors

in the
Department of Mathematics
Faculty of Arts and Sciences

June 2009

Abstract

Resilient and exposure-resilient functions are functions whose output appears random even if some portion of their input is either revealed or fixed. We explore an alternative way of characterizing these objects that ties them explicitly to the theory of randomness extractors and simplifies current proofs of basic results. We also describe the inclusions and separations governing the various classes of resilient and exposure-resilient functions. Using this knowledge, we explore the possibility of improving existing constructions of these functions and prove that one specific method of doing so is impossible.

Acknowledgments

First and foremost, I owe special thanks to my advisor, Professor Salil Vadhan, for the hours of reading and thinking that he dedicated to my work. This would have been impossible without his consistent, supportive, and insightful presence. I would also like to thank: Pablo Azar, Eleanor Birrell, Seth Flaxman, Madelyn Finucane, Mariel Finucane, Rishi Gupta, and Daniel Steinbrook for their helpful suggestions; my parents, Daniel and Shoshana, and my brother, David, for their careful proofreading and much more; Hilary Finucane, who has been showing me new intellectual horizons for almost ten years; and Reginald Hahne and Yair Schaffer, who never met each other but together nourished the curious interest of a child into the deep fascination of a student.

Contents

Chapter 1

Introduction

Suppose we roll a fair die. The number we obtain will be *random* because it will have been generated by a completely unpredictable process. That is, there is no way to predict the result of our roll more that is more accurate than an arbitrary guess. Such random (that is, completely unpredictable) numbers are desirable for lotteries, scientific simulations, online casinos, computer games, encryption schemes, and randomized clinical trials, to name just a few examples. But how can they be obtained if one does not want to spend years rolling dice?

1.1 Randomness and Pseudorandomness

Roughly a hundred years ago, the generation of long sequences of random numbers was a time-consuming task. Researchers would either use traditional mechanisms (roulette wheels, coins, etc.) or buy large tables of pre-generated random numbers like Leonard H.C. Tippet's 1927 book of 41,600 random digits extracted from census data and published by Cambridge University Press [1].

With the advent of electronic computing, researchers looked for more efficient ways of obtaining random numbers. However, getting computers to generate truly random digits is difficult because computer programs are fundamentally deterministic: given the same input, a computer program will produce the exact same output every time that it is run. This determinism is the opposite of a die's unpredictability because if we are given the input of the computer program and the program's source code, we can predict its output with complete accuracy.

Despite these limitations, algorithms were devised early on that could take a relatively short and truly random input (called a *seed*) and produce a long sequence of digits whose distribution "looked" random in a variety of ways. (For example, each digit appeared with approximately the same frequency, and there were no strings of digits that repeated themselves too often.) Though such sequences looked random in some useful ways, they had the significant drawback that they were not guaranteed to pass *any* statistical test of randomness; they were only guaranteed to pass the specific tests for which they were designed. For this reason, using them involved very real risks. In World War II, for instance, the Japanese PURPLE cipher was consistently broken by American intelligence because the passwords used for its encryption algorithm exhibited patterns that were found and exploited [2]. A similar problem struck American researchers in the 1970's when flaws were discovered in RANDU, a widely used algorithm for generating sequences that looked random, and many scientific results based on the algorithm were called into question [3, 4].

Examples such as these demonstrated early on the fundamental need for a theoretical understanding of the nature of computational randomness. This need was answered by the development of the theory of *pseudorandomness*, which seeks to understand how to efficiently generate objects that provably and quantifiably "look" random despite being constructed using little or no true randomness. Advances in this theory have yielded algorithms capable of turning a short, truly random seed into a sequence that, rather than passing a small battery of pre-specified statistical tests of randomness, can pass *any* efficiently computable statistical test of randomness with very high probability. The algorithms that produce these so-called *pseudorandom sequences* are called *pseudorandom generators* (PRG's), and their development is an active branch of the theory of pseudorandomness.

1.2 Pseudorandom Generators Versus Randomness Extractors

PRG's provide a powerful way of taking a short random seed and "stretching" it into a large amount of pseudorandomness for use in computation. However, to produce a pseudorandom sequence, a PRG still requires a truly random seed. Thus, PRG's are not the solution to our fundamental problem of how to "create" randomness for computational purposes.

To solve the problem, we must look outside the (deterministic) computer program for a source of unpredictability. There are many such sources, from the quantum phenomena of noisy diodes on the computer's chips to the timings of a user's keystrokes. The issue is that such sources of randomness, though they contain some degree of unpredictability, might exhibit significant trends (like long breaks in a user's typing followed by rapid activity) that could make them *mostly* predictable and therefore unsuitable for our purposes. However, if there were a program that converted this "imperfect" (i.e. mostly predictable) random input into "nearly perfect" (i.e. almost completely unpredictable) random output, we would have a solution to our problem. Luckily, there do exist algorithms that convert such "low-quality randomness" into "high-quality randomness", and they are called *randomness extractors*. Their study is part of the theory of pseudorandomness, and it is with an eye towards them that we will approach resilient and exposure-resilient functions.

1.3 Exposure-Resilience and the Key Exposure Problem

Cryptographic protocols are designed to allow a group of "honest" parties to perform a complex procedure—like encryption, message authentication, or identification—in spite of interference from "adversaries". A large number of these protocols depend on exploiting the fact that one or more of the honest parties possess a secret random string (called a "key"). However, standard cryptographic tools provide security only under the assumption that this key remains entirely secret from any adversary. The problem of dealing with the possibility that an adversary may discover part or all of the key is called the *key exposure problem*.

The key exposure problem is of great practical interest and represents one of the most realistic threats to security today [5]. In 1998, Nicko van Someren demonstrated a method of scanning a computer's memory for areas of high entropy and extracting confidential keys stored in that memory [6]. Only weeks after this finding was published [7], viruses emerged that exploited van Someren's idea to steal secret keys [8]. Shamir and van Someren [7] suggested methods to prevent such attacks from revealing a key in its entirety. However, securing cryptographic protocols against *partial* key exposure remained an open problem.

Exposure-resilient functions (ERF's), thus named because the secrecy of their output is "resilient" to partial exposure of their input, were introduced as a solution to this problem by Canetti et al. [9] in 2000. Intuitively, an exposure-resilient function is a function whose output appears random, even to an adversary who knows some portion of the function's input bits. More precisely, a k-ERF f is a function from n-bit strings to m-bit strings such that, for any randomly chosen n-bit string r, no adversary can tell the difference between $f(r)$ and a randomly chosen m-bit string with any significant success rate, even if the adversary knows all but k of the bits of r.

How do these functions solve the key exposure problem? If an honest party (call her Alice) possesses a randomly chosen n-bit key r along with a k-ERF f, and an adversary (call her Eve) has obtained all but k of the bits of r, Alice can simply compute $f(r)$ to obtain a new m-bit key that appears random to Eve. Like a randomness extractor, f essentially takes in "low-quality" randomness (the randomly chosen key about which Eve has some information) and deterministically produces "high-quality" randomness (the new key, about which Eve cannot predict anything).

Because they require no additional randomness in order to produce output that is effectively random (and is therefore useful for any cryptographic purpose), these functions provide a very satisfying solution to the key exposure problem in two ways. First, as we have seen, obtaining extra randomness can require significant computational resources and can be done no faster than the rate at which a computer can gather data about its surroundings. Second, suppose that Alice is collaborating with another honest party named Bob and that they both need to have the *same* randomly chosen key r for some protocol. Then when Eve learns part of r, Alice and Bob can independently apply f to r and, since f is deterministic, they will both end up with the same new key, allowing them to continue using their protocol. (This procedure is detailed in Figure 1.1.) By applying f to r, they have renewed their old key and solved the key exposure problem without having to exchange *any* new information!

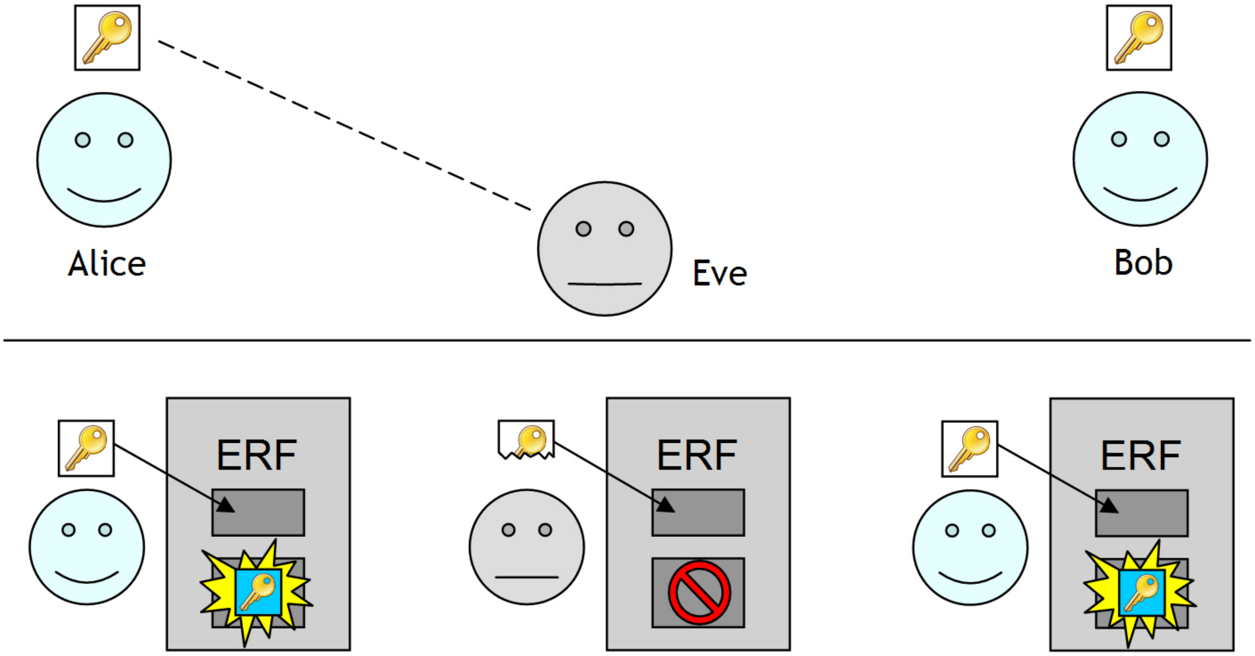

FIGURE 1.1: Alice and Bob have agreed on a shared random string (called a key), but they think that Eve may have discovered some portion of it. Alice and Bob each take this key and pass it through an ERF. They each obtain a second shared key that they can use to carry out their cryptographic protocol, and since f depends on no additional randomness, they can do this without exchanging any new information. Eve, who possesses only partial knowledge of the first key, cannot predict *anything* about the second key because of the special properties of the ERF. Note that, as this illustration suggests, the ERF is public; that is, we assume that Eve also has access to the same ERF that Alice and Bob use. This knowledge, however, does not help her.

1.4 The Ideal ERF

When designing an ERF f, we typically have three goals in mind. We want f to offer protection against as much key exposure as possible, so we seek to minimize k (the number of bits that must be protected). We also want to maximize m, the length of the output of f, since we need f to produce a new key that is long enough to be used in place of the old one. Lastly, we wish to make f's output appear as random as possible to Eve. The extent of this randomness is quantified by the probability ε that Eve will be able to predict something about $f(r)$ given her partial knowledge of r. It turns out that allowing ε to be non-zero but still keeping it very small (smaller even than the probability that Alice's computer will malfunction and compromise the cryptographic protocol without her knowledge) enables us to do significantly better in minimizing k and maximizing m, as we discuss below.

1.5 Variations on a Theme

There are many variants of the idea behind ERF's that require different degrees of randomness from f's output and assume different abilities on Eve's part to compromise the security of the original key.

For example, instead of demanding that f's output be *almost* completely unpredictable to Eve (which corresponds to having ε be very small as mentioned above) we can demand that it be *completely* unpredictable (i.e. $\varepsilon = 0$); or we could relax our requirements by saying that the output can be predictable, as long as making any prediction takes a very long time (on the order of thousands of years, for instance). Vastly different values of k and m are achievable under these different security requirements. If $\varepsilon = 0$, for example, we can prove that m cannot be greater than k, whereas if we allow a small, non-zero ε and decide that any prediction whose generation takes thousands of years is okay, we can have m polynomially larger than k.

Looking on the other hand at Eve's ability to compromise the key r, we could—instead of allowing her merely to *access* some portion of the bits of r—enable her to actively *sabotage* r by altering some of its bits. The types of functions that allow Bob to obtain random bits from Alice under these more difficult circumstances are called *resilient functions* (RF's, as opposed to ERF's) and were actually introduced by Kurosawa, Johansson, and Stinson [10] in 1997, before the introduction of ERF's.

These and many other variants of resilient and exposure-resilient functions are related in different ways—for example, it is easy to show that every resilient function can also be used as an exposure-resilient function. However, much remains to be understood both about the fundamental limitations of these functions (what parameters are achievable for different variants?) and about their relationships to each other and to other, more basic pseudorandom objects.

1.6 Contributions of This Thesis

The contributions of this work are two-fold: first, we present a new, alternate characterization of the various types of resilient and exposure-resilient functions that establishes a simple and explicit relationship with randomness extractors. From this new angle we deduce that the matter of resilience versus exposure-resilience is a distinction between worst-case and average-case behavior. Understanding the various RF's and ERF's in this way, we give an original exposition of their basic properties and the relationships among them, along with constructions and applications. It is hoped that this exposition succeeds in using our new insight to clarify and better motivate the proofs in the literature.

Second, we investigate the question of whether there is a general procedure by which, given an existing exposure-resilient function, we can somehow "amplify" its exposure-resilience property by increasing its input size n while keeping k the same (thus decreasing the percentage of the input bits that require protection). Such results have offered significant insight into the fundamental limits of achievable parameters for other pseudorandom objects in the past: for example, the discovery of a procedure to decrease the amount of randomness required by a given randomness extractor has led to extractor constructions that match upper bounds in terms of the amount of randomness that they extract. Likewise, knowing whether such a procedure exists for ERF's would offer insight into what is possible in terms of *their* parameters. In Chapter 5, we state and prove an original negative result which shows that, under certain conditions, no such method of "resilience amplification" exists for exposure-resilient functions. We also discuss the implications of this result for a current open question concerning the achievable parameters of RF's and ERF's.

1.7 Outline

We begin by stating some notational conventions and giving a brief overview of the basics of the theory of pseudorandomness that will be relevant to our work (Chapter 2). We then give an exposition of the standard definitions of the different resilient and exposure-resilient functions, along with our alternate characterization in terms of extractors and average-case behavior. Using this alternate characterization, we exhibit a few constructions and discuss applications of the various types of resilient and exposure-resilient functions (Chapter 3). We then continue our development of the theory by using our alternate characterization to completely specify the relationships between these functions (Chapter 4). Finally, we discuss the question of whether there exists a general procedure by which we can amplify the exposure-resilience properties of ERF's (Chapter 5). We conclude by discussing possibilities for further research (Chapter 6).

Chapter 2 can be safely skipped by anyone familiar with the basics of the theory of pseudorandomness. Chapters 3, 4, and 5, however, should not be skipped even by someone familiar with the literature, since each contains some degree of new material that is built upon later in the work. The exceptions to this rule are Sections 3.1 (standard definitions of RF's and ERF's), 3.3 (applications of RF's and ERF's), and 3.4 (constructions of RF's and ERF's), which may be skipped by a well-versed reader.

Chapter 2

Preliminaries

In this chapter we lay the foundations for our work. We begin by stating some notational conventions, then formalize our notion of what it means for a distribution to be "almost random". With that knowledge, we go on to discuss two of the fundamental objects of the theory of pseudorandomness: randomness extractors, which try to obtain nearly perfect randomness from imperfect random sources, and pseudorandom generators (PRG's), which take a perfectly random string of bits and stretch it into a longer string that "looks random".

2.1 The Basics

We first state some non-standard conventions that we will use when talking about sets of strings in $\{0,1\}^* = \bigcup_k \{0,1\}^k$. We then explain our notation of probability distributions. Lastly, we discuss how we will characterize asymptotic behavior, first of functions and then of algorithms.

2.1.1 Sets

We will use the convention that for any $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. Additionally, for any $w \in \{0,1\}^n$ and any set $L \subset [n]$, we will write $[w]_L$ to denote the bits of w whose positions are in L. For $i \in [n]$, we subtly abuse this notation and write $[w]_i$ to mean $[w]_{\{i\}}$. We will use $\left\{ {n \atop \ell} \right\}$ to denote the set $\{L \subseteq [n] : |L| = \ell\}$ and, given some $L \in \left\{ {n \atop \ell} \right\}$ and some $a \in \{0,1\}^\ell$, we use $L^{a,n}$ to denote the set $\{w \in \{0,1\}^n : [w]_L = a\}$. Finally, given some $L \in \left\{ {n \atop \ell} \right\}$, $a \in \{0,1\}^\ell$, and $M \subset [n]$, we will write $[L^{a,n}]_M$ to mean $\{[w]_M : w \in \{0,1\}^n, [w]_L = a\}$. When this set has only one element (that is, when $M \subset L$), then we will use $[L^{a,n}]_M$ to denote that single element rather than the set itself. For example, if $L \subset [3]$ is the set $\{1, 2\}$ then $L^{01,3} \subset \{0,1\}^3$ is the set $\{010, 011\}$ and if $M = \{1\}$ then $[L^{00,3}]_M = 0$.

2.1.2 Distributions

Throughout this work we will write distributions surrounded by angle brackets (for example: "Consider the distribution $\langle X \rangle \ldots$") to distinguish them from sets, and we will write $w \leftarrow \langle X \rangle$ to refer to a string w chosen according to the distribution $\langle X \rangle$. When the support of a distribution $\langle X \rangle$ contains only one element, we will say that $\langle X \rangle$ is a *degenerate distribution.* If a distribution is equally likely to yield any element of its support when sampled, we will refer to it as *flat*, or say that it is a *uniform* distribution.

Flat distributions are so important in our work that we will adopt the convention that if $\langle X \rangle$ is a flat distribution then X is implicitly defined to be its support and, likewise, if X is a set then $\langle X \rangle$ is implicitly defined to be the uniform distribution over X. For example, for every $L \in \left\{ {n \atop \ell} \right\}$ and $a \in \{0,1\}^\ell$, $\langle L^{a,n} \rangle$ is the uniform distribution on $L^{a,n}$, and if we define $\langle A \rangle := \langle L^{a,n} \rangle$, then $A = L^{a,n}$. Note that this convention would cause ambiguity if we did not restrict its application to flat distributions only, because if for some non-flat $\langle X \rangle$ we call its support X and then implicitly define $\langle X \rangle$ to be the uniform distribution over X we have implicitly re-defined $\langle X \rangle$. For this reason, the convention applies *only* if $\langle X \rangle$ is flat, and we will use it only when a distribution clearly satisfies this condition. We will also ignore this convention when referring to the uniform distribution over $\{0,1\}^n$—instead of writing it as $\langle \{0,1\}^n \rangle$, we will simply write $\langle U_n \rangle$.

Given two distributions $\langle X \rangle$ and $\langle Y \rangle$ and some constant $\gamma \in [0,1]$, we use $\gamma \langle X \rangle + (1-\gamma)\langle Y \rangle$ to denote the distribution obtained by sampling $\langle X \rangle$ with probability γ and sampling $\langle Y \rangle$ with probability $(1-\gamma)$. If $\langle X \rangle$ and $\langle Y \rangle$ are jointly distributed (that is, if they are functions of the same underlying random experiment), then the joint distribution of $\langle X \rangle$ and $\langle Y \rangle$ is denoted by $(\langle X \rangle, \langle Y \rangle)$.

We think of distributions as random variables in that we write $\langle X \rangle|_{\langle Y \rangle = y}$ to refer to the random variable we obtain by sampling $\langle X \rangle$ conditioned on $\langle Y \rangle = y$. Additionally, for some function f on $\{0,1\}^n$ and some distribution $\langle X \rangle$ over $\{0,1\}^n$, $f(\langle X \rangle)$ is the random variable sampled by choosing some $w \leftarrow \langle X \rangle$ and calculating $f(w)$. If we write the same distribution more than once inside parentheses then we are referring to the same random variable in both cases, not independent identically distributed copies of that random variable. If we really want two independent copies of a random variable, we will distinguish between them by adding a $'$ to one of them. For example, a string chosen from the distribution $(\langle U_n \rangle, \langle U_n \rangle)$ will always be of the form $w \circ w$ for some $w \in \{0,1\}^n$, whereas the distribution $(\langle U_n \rangle, \langle U'_n \rangle)$ is equal to $\langle U_{2n} \rangle$.

We will denote expectation with $\mathrm{E}_{w \leftarrow \langle X \rangle}[\cdots]$ and use $\Pr[\cdots]$ to refer to the probability of an event taking place. In any expression of the form $\Pr[\cdots]$, the probability is always taken over *all* random variables in the $[\cdots]$ unless otherwise noted.

2.1.3 "Big O" Notation

In the interest of not abusing notation, we will try to treat uses of "Big O" notation as if they refer to sets (for example: $f \in O(g)$). However, when this is cumbersome we will occasionally use $f = O(g)$ or $f \leq O(g)$ to mean the same thing. This comes up especially in cases when we want to say, for instance, that f is bounded by $g + h$ where $h \in O(g')$. In such a situation, we will simply write $f \leq g + O(g')$.

2.1.4 Algorithms and Polynomial Time

For a randomized algorithm A, we will separate the part of the input chosen deterministically from the part chosen at random by writing $A(x; r)$ where x is the user-chosen input and r is a string of bits chosen uniformly at random.

When we say that a function f is polynomial-time computable, we mean that there is a uniform algorithm that can compute f on inputs of arbitrarily long length in polynomial time. More formally, if we say, for some countable $N \subset \mathbb{N}$, that "for all $n \in N$ there exists a polynomial-time computable function $f \colon \{0,1\}^n \to \{0,1\}^m$ satisfying the property P", we mean that there exists an infinite family of functions $\{f_n\}_{n \in N}$ where each f_n is a function on $\{0,1\}^n$ and has the property P, together with a uniform algorithm A which, when given a string w of length $n \in N$, can calculate $f_n(w)$ in time asymptotically bounded by a polynomial in n.

2.2 Indistinguishability

Throughout this work, we will need to quantify how "random" various objects are. To answer this question, we simply need a precise way of measuring how close to uniform a given distribution is—the closer it is, the more "random" it is. There are a few different ways to define this notion of closeness, but all can be understood through the lens of indistinguishability.

2.2.1 Perfect Indistinguishability

It is easy to state the strongest possible formulation of statistical "closeness". Two distributions $\langle X \rangle$ and $\langle Y \rangle$ are as close as possible to each other if they are identically distributed. In this case we say that $\langle X \rangle$ and $\langle Y \rangle$ are *perfectly indistinguishable.*

Definition 2.1 (Perfectly Indistinguishable Distributions)**.** Two distributions $\langle X \rangle$ and $\langle Y \rangle$ on a set S are *perfectly indistinguishable* if and only if for every $w \in S$ we have $\Pr[\langle X \rangle = w] = \Pr[\langle Y \rangle = w]$.

In this work, we will write $\langle X \rangle = \langle Y \rangle$ to denote that $\langle X \rangle$ and $\langle Y \rangle$ are perfectly indistinguishable. But why call the two distributions "indistinguishable" rather than simply calling them equal? The reason is that this definition can also be arrived at by imagining a scenario in which an algorithm A, given an input z, is trying to decide whether z was drawn from $\langle X\rangle$ or $\langle Y\rangle$ by outputting 1 if it thinks z came from $\langle X\rangle$ and outputting 0 otherwise. If $\langle X\rangle$ and $\langle Y\rangle$ are identically distributed then A will perform identically regardless of which distribution z came from and so it will have no chance of figuring this out regardless of the computational resources (random bits, time, program length, etc.) at its disposal. Thus, we can write that $\langle X\rangle$ and $\langle Y\rangle$ are perfectly indistinguishable if and only if, for every computationally unbounded algorithm A we have that

$$\Pr\left[A(\langle X\rangle) = 1\right] = \Pr\left[A(\langle Y\rangle) = 1\right]$$

2.2.2 Statistical Indistinguishability

The requirement that $\langle X\rangle = \langle Y\rangle$ is quite strong, and the distributions that we discuss will rarely have the property that no algorithm at all can *ever* distinguish them from uniform. However, relaxing this notion a bit by requiring our distributions to *almost* always fool a computationally unbounded distinguisher makes many impossible tasks possible. In the following definition, our computationally unbounded distinguisher A is modelled by specifying the set T of inputs on which it outputs 1.

Definition 2.2 (Statistical Distance)**.** Let $\langle X\rangle$ and $\langle Y\rangle$ be any two distributions on a set S. The *statistical distance* $\Delta(\langle X\rangle, \langle Y\rangle)$ between $\langle X\rangle$ and $\langle Y\rangle$ is defined by

$$\Delta(\langle X\rangle, \langle Y\rangle) = \max_{T\subset S} |\Pr[\langle X\rangle \in T] - \Pr[\langle Y\rangle \in T]|$$

By modeling the distinguisher A using one set, we have implicitly restricted ourselves to considering only non-randomized distinguishers. We will soon see that this loses us no generality. In the meantime, though, we will want a way to talk specifically about the ability of various algorithms to distinguish between two distributions.

Definition 2.3 (Advantage)**.** Let $\langle X\rangle$ and $\langle Y\rangle$ be any two distributions on a set S. Let A be a (possibly randomized) algorithm, taking inputs in S, that outputs either 0 or 1. If

$$|\Pr[A(\langle X\rangle) = 1] - \Pr[A(\langle Y\rangle) = 1]| = \varepsilon$$

(where the probability is taken over $\langle X\rangle$, $\langle Y\rangle$, and any potential coin flips of A), then we say that A has *advantage* ε in distinguishing $\langle X\rangle$ from $\langle Y\rangle$.

Using this new terminology, we can say that $\Delta(\langle X\rangle, \langle Y\rangle) = \varepsilon$ means that there exists a non-randomized algorithm that has advantage ε in distinguishing between the two distributions and that no non-randomized algorithm can do better. We now show that in fact this characterization holds even when we consider distinguishers with access to perfectly random bits.

Proposition 2.4. *Let $\langle X\rangle$ and $\langle Y\rangle$ be any two distributions on a set S. Then $\Delta(\langle X\rangle,\langle Y\rangle)\geq\varepsilon$ if and only if there exists a randomized algorithm A taking inputs in S and outputting either 0 or 1 that has advantage ε in distinguishing $\langle X\rangle$ from $\langle Y\rangle$.*

Proof. The direction $\Rightarrow$ is clear from the above explanation. The reason that $\Leftarrow$ holds is that if we have a randomized algorithm $A(x;r)$ on S that has advantage ε in distinguishing $\langle X\rangle$ from $\langle Y\rangle$, then there is at least one setting r_0 of A's random bits on which it must have advantage at least ε. If we fix r to r_0, then we have a deterministic algorithm with advantage at least ε and we can take T to be the set of inputs on which that algorithm accepts. □

We will write $\langle X\rangle\cong_\varepsilon\langle Y\rangle$ to denote that $\langle X\rangle$ and $\langle Y\rangle$ satisfy $\Delta(\langle X\rangle,\langle Y\rangle)\leq\varepsilon$. Since, as Proposition 2.4 shows, our definition of statistical distance captures the abilities even of randomized distinguishers, we have that $\langle X\rangle=\langle Y\rangle$ if and only if $\langle X\rangle\cong_0\langle Y\rangle$.

There are two interesting (and useful) equivalent ways to define statistical distance if we view distributions over a set S as $|S|$-dimensional vectors with entries in $[0,1]$. We state them in the following proposition, the proof of which is omitted.

Proposition 2.5. *Let $\langle X\rangle$ and $\langle Y\rangle$ be any two distributions on a set S, and define*

$$A=\{w\in S:\ \Pr[\langle X\rangle=w]>\Pr[\langle Y\rangle=w]\}$$

Then we have that

$$\Delta(\langle X\rangle,\langle Y\rangle)=\sum_{w\in A}\left(\Pr\left[\langle X\rangle=w\right]-\Pr\left[\langle Y\rangle=w\right]\right)=\frac{1}{2}\sum_{w\in S}\left|\Pr\left[\langle X\rangle=w\right]-\Pr\left[\langle Y\rangle=w\right]\right|$$

These reformulations allow us to prove some basic facts about Δ.

Lemma 2.6. *For every two distributions $\langle X\rangle$ and $\langle Y\rangle$, Δ satisfies the following natural properties.*

1. *Symmetry: $\Delta(\langle X\rangle,\langle Y\rangle)=\Delta(\langle Y\rangle,\langle X\rangle)$*
2. *Triangle inequality: For any third distribution $\langle Z\rangle$, we have*

$$\Delta(\langle X\rangle,\langle Y\rangle)\leq\Delta(\langle X\rangle,\langle Z\rangle)+\Delta(\langle Z\rangle,\langle Y\rangle)$$

3. *Positive definiteness and boundedness: $0\leq\Delta(\langle X\rangle,\langle Y\rangle)\leq 1$ with equality for identical distributions and distributions with disjoint support respectively.*
4. *For any function f on S, we have $\Delta(f(\langle X\rangle),f(\langle Y\rangle))\leq\Delta(\langle X\rangle,\langle Y\rangle)$*
5. *For any third distribution $\langle Y'\rangle$ and constant $\gamma\in[0,1]$, we have*

$$\Delta(\langle X\rangle,\gamma\langle Y\rangle+(1-\gamma)\langle Y'\rangle)\leq\gamma\Delta(\langle X\rangle,\langle Y\rangle)+(1-\gamma)\Delta(\langle X\rangle,\langle Y'\rangle)$$

6. *For any third distribution $\langle Y' \rangle$, we have*

$$\Delta\left((\langle X\rangle, \langle Y\rangle), (\langle X\rangle, \langle Y'\rangle)\right) = \mathop{\mathrm{E}}_{w \leftarrow \langle X\rangle}\left[\Delta\left(\langle Y\rangle|_{\langle X\rangle = x}, \langle Y'\rangle|_{\langle X\rangle = x}\right)\right]$$

Proof. Looking to the second alternate definition of Δ in Proposition 2.5, we see that if we simply treat $\langle X\rangle$ and $\langle Y\rangle$ as functions from S to $[0,1]$, then $\Delta(\langle X\rangle, \langle Y\rangle)$ is $1/2$ times the L^1-norm between the two functions. This proves properties (1), (2), and positive definiteness. Boundedness, along with (5) and (6), is shown by simple manipulations of this L^1-norm characterization of Δ.

To see (4) we observe that for any subset $T \subset S$, $|\Pr[f(\langle X\rangle) \in T] - \Pr[f(\langle Y\rangle) \in T]|$ is at most $|\Pr[\langle X\rangle \in f^{-1}(T)] - \Pr[\langle Y\rangle \in f^{-1}(T)]|$, which is, in turn, at most $\Delta(\langle X\rangle, \langle Y\rangle)$. □

2.2.3 Computational Indistinguishability

In the study of pseudorandomness we sometimes find even guarantees of statistical closeness to be too stringent. This is because, as Proposition 2.4 shows, statistical closeness requires that any algorithm—even a computationally *unbounded* one—be unable to really distinguish between the two distributions in question. Since we often seek this kind of security only against *efficient* algorithms, the natural thing to do is to restrict the run-time of the algorithm A in Proposition 2.4 and call the resulting definition "computational indistinguishability". We run into a slight complication with this approach though: if the two distributions being distinguished have finite supports contained in $\{0,1\}^*$ then any adversary can simply hardwire in a lookup table and decrease its run-time to below whatever efficiency restriction we have imposed, thus reducing our new definition to simple statistical distance. This is, of course, the same difficulty that we find when defining what it means for an algorithm to run in polynomial time, and we fix it in the same way: by talking asymptotically, which in this case requires us to make an additional definition.

Definition 2.7 (Ensemble)**.** An *ensemble* is a family of distributions $\{\langle X_n\rangle\}_{n\in N}$ where $N \subset \mathbb{N}$ is countable and $\langle X_n\rangle$ is a distribution over $\{0,1\}^n$. An ensemble $\{\langle X_n\rangle\}_{n\in N}$ is said to be *polynomial-time constructible* if and only if there exists a randomized algorithm A, requiring $d(n)$ random bits on inputs of length n, such that $A(1^n; \langle U_{d(n)}\rangle) = \langle X_n\rangle$ and A runs in time asymptotically bounded by a polynomial in n.

Now we can state the definition of computational indistinguishability that we wanted in the first place.

Definition 2.8 (Computational Indistinguishability for Ensembles)**.** Two ensembles $\{\langle X_n\rangle\}_{n\in N}$ and $\{\langle Y_n\rangle\}_{n\in N}$ are $(t(n), \varepsilon(n))$-computationally indistinguishable if and only if every randomized, non-uniform algorithm A running in time asymptotically bounded by $t(n)$ has advantage at most $\varepsilon(n)$ in distinguishing $\langle X_n\rangle$ from $\langle Y_n\rangle$ for $n \in N$ sufficiently large.[1]

[1] Note that we have defined computational indistinguishability in terms of randomized, non-uniform adversaries. There are deterministic and uniform analogues of this definition, but we will not use them in this work since all our results hold for this stronger definition.

We express computational indistinguishability using the symbol $\cong^c$. That is, the expression $\{\langle X_n\rangle\}_{n\in N} \cong^c_{\varepsilon(n)} \{\langle Y_n\rangle\}_{n\in N}$ means that for every polynomial $p(n)$ we have that $\{\langle X_n\rangle\}_{n\in N}$ and $\{\langle Y_n\rangle\}_{n\in N}$ are $(p(n), \varepsilon(n))$-computationally indistinguishable.

As we do with polynomial-time computable functions, we will often suppress the parametrization by n when talking about computational indistinguishability, and so we will frequently use $\langle X\rangle \cong^c_\varepsilon \langle Y\rangle$ instead of the more cumbersome $\{\langle X_n\rangle\}_{n\in N} \cong^c_{\varepsilon(n)} \{\langle Y_n\rangle\}_{n\in N}$.

At this point we've defined three relations on ensembles of distributions: $=$, $\cong_\varepsilon$, and $\cong^c_\varepsilon$. Sometimes, though, we will want to make a statement that applies to all three relations or for which the choice of relation is clear from context. In these cases we will use $\approx$.

2.3 Randomness Extraction

We now turn to the following problem: Given a coin that is biased in some known way (say it has probability $p \neq \frac{1}{2}$ of landing on heads), is there a way to use the coin to simulate an unbiased one? This question is a special case of one of the central problems in the theory of pseudorandomness: the simulation of perfectly random bits using some sort of imperfect random source. Randomness extractors, the objects that perform this task, have unified many seemingly disparate aspects of the theory, and we employ them to that end in this work as well. Before we can really talk about randomness extractors though, we must decide what we mean when we talk about having an "imperfect random source".

2.3.1 Random Sources

The biased coin is one example of an imperfect random source. To discuss it more formally, we can define a *class of sources* that represents independent flips of biased coins. These are called *Von Neumann Sources.*

Definition 2.9 (Von Neumann Sources)**.** A distribution $\langle X\rangle$ over $\{0,1\}^n$ is an (n, p) Von Neumann source if and only if $\langle X\rangle = (\langle X_1\rangle, \langle X_2\rangle, \ldots, \langle X_n\rangle)$ where each $\langle X_i\rangle$ is a random bit and we have $\Pr[\langle X_i\rangle = 0] = p$ for every i.

Using this language, we can now state that the task of simulating an unbiased coin given a biased one with unknown bias is equivalent to the task of building a function that extracts good randomness from a Von Neumann source on n bits (regardless of p). A simple construction of a function that extracts m bits that are statistically close to uniform from *any* Von Neumann source on n bits exists and is left as an exercise. Von Neumann sources are not very useful, however, because the assumptions they require us to make about both the independence between the bits and their identical nature are quite strong. Ideally, if we want to extract m random bits from a source, we'd like to assume only that the source has m "bits of randomness" hiding in it somehow. The concept of min-entropy, described below, helps us to do this.

Definition 2.10 (Min-entropy[2])**.** Let $\langle X \rangle$ be a distribution on a set S. The *min-entropy* of $\langle X \rangle$ is defined by

$$H_\infty(\langle X \rangle) = \min_{w \in S} \left\{ \log \frac{1}{\Pr[\langle X \rangle = w]} \right\}$$

(The logarithm in this definition, like all subsequent logarithms in this work, is to the base 2.)

We observe that H_∞ has the following properties:

- $0 \leq H_\infty(\langle X \rangle) \leq \log |S|$ with equality when $\langle X \rangle$ is constant and when $\langle X \rangle$ is the uniform distribution on S respectively.
- If $\langle X \rangle$ and $\langle Y \rangle$ are independent, then $H_\infty((\langle X \rangle, \langle Y \rangle)) = H_\infty(\langle X \rangle) + H_\infty(\langle Y \rangle)$.
- For every deterministic function f, we have $H_\infty(f(\langle X \rangle)) \leq H_\infty(\langle X \rangle)$.

We are now in a position to define a class of weak random sources from which it would be useful to extract randomness.

Definition 2.11 (Min-Entropy Source)**.** A distribution $\langle X \rangle$ on $\{0,1\}^n$ is a *k-min-entropy source* (usually called a *k-source*) if and only if $H_\infty(\langle X \rangle) \geq k$

We can think a k-source $\langle X \rangle$ as a source in which the probability of drawing any one value upon sampling $\langle X \rangle$ is at most 2^{-k}, because $H_\infty(\langle X \rangle) \geq k$ if and only if $\Pr[\langle X \rangle = w] \leq 2^{-k}$ for all $w \in S$.

For most of this work, we will concern ourselves with one specific sub-class of the class of k-sources on n bits: those k-sources consisting of k uniformly random and independent bits and $n-k$ fixed bits. Because the fixed bits do not depend on the random bits, these are called *oblivious bit-fixing sources* (OBFS's); we formally define them below.

Definition 2.12 (Oblivious Bit-Fixing Sources)**.** A distribution $\langle X \rangle$ on $\{0,1\}^n$ is an (n,k) *oblivious bit-fixing source* if and only if for some $L \in \left\{ {n \atop n-k} \right\}$ and some $a \in \{0,1\}^{n-k}$, we have $\langle X \rangle = \langle L^{a,n} \rangle$

2.3.2 Extractors

With a precisely defined class of sources in mind, together with the notion of statistical distance, it is easy to define what we want from a randomness extractor. We simply want it to take any source from the class of sources and turn it into a distribution that is statistically close to the uniform distribution. However, as we will see, this turns out to be impossible to do in general without access to a small number of perfectly random bits. This has led to the introduction of two different types of extractors.

[2] The reason for the notation H_∞ is that there is a whole family of entropy functions $\{H_c : c \in \mathbb{R}_{\geq 0}\}$ (H_c is called the *Renyi entropy of order* c), and $H_\infty = \lim_{c \to \infty} H_c$.

Deterministic Extractors

Deterministic extractors are the simplest type of extractors that we consider, and they are the most limited in terms of the classes of sources that they can handle.

Definition 2.13 (Deterministic Randomness Extractor)**.** Let $\mathcal{C}$ be a class of sources on $\{0,1\}^n$. A *deterministic ε-extractor for $\mathcal{C}$* is a function $E\colon \{0,1\}^n \to \{0,1\}^m$ such that for every $\langle X\rangle \in \mathcal{C}$ we have $E(\langle X\rangle) \cong_\varepsilon \langle U_m\rangle$.

Note that we require that there be *one* function E that works for *all* sources in $\mathcal{C}$. This captures the notion that though we may know, for example, that our source is a biased coin, we may not know what the coin's particular bias is when we are designing E. Similarly, though we know that some (n,k) oblivious bit-fixing source $\langle X\rangle$ will have $n-k$ fixed bits, we may not know ahead of time which bits those are or the values to which they will be fixed.

Unfortunately, it is easy to show that there is no hope of building a deterministic extractor for general k-sources.

Proposition 2.14. *For every function $E\colon \{0,1\}^n \to \{0,1\}$, there exists an $(n-1)$-min-entropy source $\langle X\rangle$ such that $E(\langle X\rangle)$ is a degenerate distribution.*[3]

Proof. There exists some $b \in \{0,1\}$ such that $|E^{-1}(\{b\})| \geq 2^n/2$. Let $\langle X\rangle$ be the uniform distribution on $E^{-1}(\{b\})$. □

This result shows that even if we assume that a k-source has almost as much entropy as it can without being uniform, we will not be able to reliably extract even *one* bit from it. If we restrict our attention to smaller classes of sources though, then extraction of randomness does become possible.

There is quite an array of deterministic extractors for various types of sources and applications in the literature. A review of these is beyond the scope of this work, but we do say a few words on deterministic extractors for oblivious bit-fixing sources since we will require them in subsequent chapters. It is easy to see that we can always extract one perfectly random bit from a $(n,1)$ oblivious bit-fixing source (by xor-ing all the bits of the input), and that we can always extract $n-1$ perfectly random bits from an $(n,n-1)$ oblivious bit-fixing source (by xor-ing successive pairs of the input bits). OBFS's with more than 1 and fewer than $n-1$ fixed bits are much harder to extract from, but there are constructions of extractors that manage to do this, one of which we state below. This construction, due to Gabizon, Raz, and Shaltiel [11], is useful only for $k \gg \sqrt{n}$.

Theorem 2.15 (Gabizon, Raz, and Shaltiel [11])**.** *For every constant $0 < \gamma < 1/2$ and for every n and k satisfying $n \geq k > n^{1/2+\gamma}$, there exists a polynomial-time computable deterministic ε-extractor for (n,k) oblivious bit-fixing sources $E\colon \{0,1\}^n \to \{0,1\}^m$ with $m \geq k - n^{1/2+\gamma}$ and $\varepsilon \leq 2^{-\Omega(n^\gamma)}$ for n sufficiently large.*

[3] Recall that a degenerate distribution is one whose support contains only one element.

In the same paper, a similar construction is given that can handle $k > (\log n)^c$ for some universal constant c, but with worse error.

Theorem 2.16 (Gabizon, Raz, and Shaltiel [11])**.** *There exist universal constants $c > 0$ and $0 < \mu, \nu < 1$ such that for all n and k satisfying $n \geq k \geq \log^c n$ there exists a polynomial-time computable deterministic ε-extractor for (n, k) oblivious bit-fixing sources $E\colon \{0,1\}^n \to \{0,1\}^m$ with $m \geq k - O(k^\nu)$ and $\varepsilon \in O(1/k^\mu)$ for n sufficiently large.*

These two constructions are based on applying a method of improving the output length of an existing extractor to two extractors of Kamp and Zuckerman [12]. In order to preserve the continuity of the exposition in the coming chapters, we delay sketches of those constructions for now; however, we will cite them here.

The first construction is the one used by Gabizon, Raz, and Shaltiel [11] to obtain Theorem 2.15. As with that construction, this one is only useful when $k \gg \sqrt{n}$.

Theorem 2.17 (Kamp and Zuckerman [12])**.** *For every constant $0 < \gamma \leq 1/2$ and every constant $c > 0$, and for every n and k satisfying $n \geq k \geq n^{1/2+\gamma}$, there exists a polynomial-time computable deterministic ε-extractor for (n, k) oblivious bit-fixing sources $E\colon \{0,1\}^n \to \{0,1\}^m$ with $m \in \Omega(n^{2\gamma})$ and $\varepsilon \leq 2^{-cm}$.*

The second construction is the one used to obtain Theorem 2.16. It works for all settings of k, but pays a price in its output length.

Theorem 2.18 (Kamp and Zuckerman [12])**.** *For every k and n satisfying $n \geq k > 0$, there exists a polynomial-time computable deterministic ε-extractor for (n, k) oblivious bit-fixing sources $E\colon \{0,1\}^n \to \{0,1\}^m$ with $m \geq \log k/4$ and $\varepsilon \leq 2^{-\sqrt{k}}$.*

Seeded Extractors

Though a deterministic extractor cannot extract randomness from k-sources in general, it turns out that if we give it access to a small number of truly random bits then extraction from such sources does become possible. The truly random bits are called the *seed* of the extractor, and so we refer to this type of extractor as a *seeded extractor*. The original motivation for the introduction of seeded extractors was the simulation of randomized algorithms, but they have acquired many other applications, one of which is the use of special seeded extractors called *strong* extractors (see below) to construct certain types of exposure-resilient functions.

Definition 2.19 (Seeded Randomness Extractor[4])**.** Let $\mathcal{C}$ be a class of sources on $\{0,1\}^n$. A *seeded ε-extractor for $\mathcal{C}$* is a function $E\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ such that for every $\langle X \rangle \in \mathcal{C}$, we have $E(\langle U_s \rangle, \langle X \rangle) \cong_\varepsilon \langle U_m \rangle$.

[4] In the literature, the bits of the seed are typically the second parameter of the function.

There is a long line of research focusing on the construction of polynomial-time computable seeded extractors with small seed length and large output length. However, the details of these constructions are inessential to our work, so we refer the reader to [13–15] for surveys on applications of seeded extractors and to [16] for a survey focusing on explicit constructions of seeded extractors.

There are cases, like the construction of exposure-resilient functions, in which we need the seeded extractor to generate output that looks random even if the value of the seed is known to an adversary. This gives rise to the notion of a strong extractor, which is a seeded extractor that includes its seed as part of its output.

Definition 2.20 (Strong Randomness Extractor)**.** Let $\mathcal{C}$ be a class of sources on $\{0,1\}^n$. A *strong ε-extractor for $\mathcal{C}$* is a function $H\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ such that for every $\langle X\rangle \in \mathcal{C}$, we have $(\langle U_s\rangle, H(\langle U_s\rangle, \langle X\rangle)) \cong_\varepsilon (\langle U_s\rangle, \langle U_m\rangle)$.

The current "state of the art" in polynomial-time computable strong extractors is a construction of Guruswami, Umans, and Vadhan, the parameters of which are given below.

Theorem 2.21 (Guruswami, Umans, Vadhan [17])**.** *For every constant $0 < \gamma < 1$, all positive integers n and ℓ, and for all $\varepsilon > 0$, there is a polynomial-time computable strong ε-extractor for ℓ-min-entropy sources $h\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ with $m \geq \gamma\ell$ and $s \leq \log n + O(\log(\ell/\varepsilon))$.*

2.4 Pseudorandom Generators

The last type of object that we introduce in this chapter is the pseudorandom generator (PRG). Intuitively, PRG's are meant to solve the following problem: given n uniformly distributed random bits, is there a way to "stretch" those bits into $\ell(n)$ bits that are close to random (with $\ell(n) > n$) by some deterministic procedure? In other words, is there a function $G\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ such that $G(\langle U_n\rangle)$ is close to $\langle U_{\ell(n)}\rangle$? Of course, even for $\ell(n) = n+1$ we cannot hope to obtain statistical closeness better than $\varepsilon = 1/2$ because the size of the support of $G(\langle U_n\rangle)$ would be half the size of the support of $\langle U_{\ell(n)}\rangle$. (Another way to think about this is that $H_\infty(G(\langle U_n\rangle)) \leq H_\infty(\langle U_n\rangle) = n$.) Therefore we only ask for *computational* indistinguishability between $G(\langle U_n\rangle)$ and $\langle U_{\ell(n)}\rangle$.

Definition 2.22 (Pseudorandom Generator)**.** A function $G\colon \{0,1\}^n \to \{0,1\}^{\ell(n)}$ is an ε-pseudorandom generator if and only if $G(\langle U_n\rangle) \cong^c_\varepsilon \langle U_{\ell(n)}\rangle$

(We are suppressing a parametrization here: this definition is actually talking about a family of functions G_n for infinitely many n, and the computational indistinguishability is between the ensembles $\{G_n(\langle U_n\rangle)\}$ and $\{\langle U_{\ell(n)}\rangle\}$)

Since we will be talking about ensembles, we will want to quantify how small the error parameter ε should be in terms of n.

Definition 2.23 (Negligibility)**.** A function $\varepsilon\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible in n if and only if, for every constant $d > 0$ we have $\varepsilon(n) < 1/n^d$ for sufficiently large n.

Pseudorandom generators, since they involve guarantees against the capabilities of all polynomial-time algorithms, naturally cannot be constructed without certain (unproven) assumptions about the limitations of such algorithms. To make clear the hardness assumption that we will use when talking about PRG's, we need to define what it means for a function to be "difficult to invert".

Definition 2.24 (One-Way Function Family)**.** A polynomial-time computable family of functions $\mathcal{F} = \{f_n \colon \{0,1\}^n \to \{0,1\}^{\ell(n)} : n \in \mathbb{N}\}$ is a *one-way function family* if and only if for every randomized, non-uniform algorithm A running in time polynomial in n, and for every $n \in \mathbb{N}$:

$$\Pr\left[A(f(\langle U_n \rangle)) \in f^{-1}(f(\langle U_n \rangle))\right] \leq \varepsilon(n)$$

for some $\varepsilon(n)$ negligible in n.

One-way functions can be made to yield PRG's, but the process is quite involved so we only cite it here.

Theorem 2.25 (Hastad et al. [18])**.** *The following are equivalent:*

1. *There exists a one-way function family.*
2. *There exists, for every $n \in \mathbb{N}$, a polynomial-time computable ε-pseudorandom generator $G \colon \{0,1\}^n \to \{0,1\}^{n+1}$ with ε negligible in n.*
3. *For every constant $c > 0$, there exists, for every $n \in \mathbb{N}$, a polynomial-time computable ε-pseudorandom generator $G \colon \{0,1\}^n \to \{0,1\}^{n^c}$ with ε negligible in n.*

Theorem 2.25 successfully turns a one-way function into a PRG that can stretch n bits into n^c bits for any constant $c > 0$, regardless of whether the one-way function stretched its input at all! This is quite powerful, as we will see when we use PRG's in some of our constructions.

Like extractors, PRG's are immensely useful and have found numerous applications. For example, they provide a natural analogue to one-time pad encryption: ordinarily, to encrypt a message x we might choose some string $r \leftarrow \langle U_{|x|} \rangle$ as a secret key and then transmit $x \oplus r$, but this requires as many random bits in our key as there are bits in our message. With a PRG, we can choose n random bits and then encrypt a message of length $\ell(n)$ by choosing $r \leftarrow \langle U_n \rangle$ and sending $x \oplus G(r)$. If $\ell(n) \gg n$, this saves us a great deal of randomness and private communication.

As with many pseudorandom objects, PRG's can be defined to fool different classes of distinguishers and to satisfy different efficiency requirements. The specific definition that we have given, with the added requirement that ε be negligible in the output length of the PRG (as it is in Theorem 2.25), yields so-called *cryptographic PRG's*. However, other variations are useful for other purposes: for instance, PRG's that work against non-uniform adversaries that run in time $t(\ell(n))$ for one *fixed* function $t(\cdot)$ can be used to de-randomize randomized algorithms running in time $t(\cdot)$. (In such applications, $\ell(n)$ and the run-time of the PRG determine the efficiency of the de-randomization.) For a general survey of the different types of PRG's that includes various constructions and further discussion of their impressive properties (including this one), see [19].

2.5 References

In the interest of leaving the flow of ideas uninterrupted, each of the remaining chapters in this work will contain a "References" section at its conclusion that will cite any sources responsible for proofs or ideas in that chapter. All lemmas/propositions/theorems stated without proof will carry a citation next to their statement in the text itself, and all lemmas/propositions/proofs for which a proof is presented will have their degree of originality (or lack thereof) explained in the final "References" section, along with the citations of any relevant sources.

The presentation in this chapter borrows liberally from [20].

Chapter 3

The RF and ERF Zoo

This chapter presents a brief survey of the basic theory of resilient and exposure-resilient functions. We will first give definitions of the objects we are examining, then give an alternate characterization of RF's and ERF's that ties them closely to the theory of randomness extractors and will serve us later in proving relationships among them. After laying this groundwork, we will mention some applications and look at a few constructions, showing in the process that some RF's and ERF's are equivalent to error-correcting codes and that the existence of some RF's and ERF's is equivalent to the existence of pseudorandom generators.

3.1 Definitions

3.1.1 Resilient Functions

Let us begin by defining the most straightforward type of function with which we will work: the *perfect resilient function*.[1]

Definition 3.1 (Perfect Resilient Function)**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a *perfect k-RF* if and only if for every $L \in \left\{ {n \atop n-k} \right\}$ and $a \in \{0,1\}^{n-k}$ we have $f(\langle L^{a,n} \rangle) = \langle U_m \rangle$.[2]

Another way to approach the same definition is to set up a game in which we have a computationally unbounded adversary A that carries out the following steps:

1. *Setup Step*: A chooses a set $L \in \left\{ {n \atop n-k} \right\}$ and a string $a \in \{0,1\}^{n-k}$ and requests that any input passed to f have the bits in L set to a.

[1] In the literature these are typically referred to simply as Resilient Functions

[2] Recall the following notational conventions: $\left\{ {n \atop n-k} \right\}$ is $\{S \subseteq [n] : |S| = n-k\}$; for $L \in \left\{ {n \atop n-k} \right\}$ and $a \in \{0,1\}^{n-k}$, $L^{a,n}$ is the set of strings $w \in \{0,1\}^n$ with $[w]_L = a$; and $\langle L^{a,n} \rangle$ is the uniform distribution on $L^{a,n}$.

2. *Query Step*: A string r is drawn from $\langle L^{a,n} \rangle$, and A receives a string z that is either $f(r)$ or a string drawn randomly from $\langle U_m \rangle$.

3. *Distinguishing Step*: A attempts to output 0 if z came from $\langle U_m \rangle$ and 1 otherwise.

Within this framework, we can say that f is a k-RF if and only if no A can ever gain any advantage over f; that is, if $|\Pr[A(\langle L^{a,n} \rangle) = 1] - \Pr[A(\langle U_m \rangle) = 1]| = 0$ for all A, L, and a.

It is easy to construct a trivial perfect 1-RF: outputting the exclusive-or of the input bits will always yield a perfectly random bit even if $n-1$ of the input bits are fixed. However, this definition is very stringent. In what ways can we relax it? One simple change we can make is to say that it is alright if some adversaries are able to gain a small advantage over f. We do this by saying that we only require $|\Pr[A(f(\langle L^{a,n} \rangle)) = 1] - \Pr[A(\langle U_m \rangle) = 1]| \leq \varepsilon$ for some ε that is negligible in an appropriate security parameter. This leads us to the following definition:

Definition 3.2 (Statistical Resilient Function)**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a *statistical* (k, ε)*-RF* if and only if for every $L \in \left\{ {n \atop n-k} \right\}$ and $a \in \{0,1\}^{n-k}$ we have $f(\langle L^{a,n} \rangle) \cong_\varepsilon \langle U_m \rangle$.[3]

We can continue to relax our demands on f by stipulating that, in addition to the fact that we do not mind if adversaries gain a small advantage over f, we also only care about adversaries that run in polynomial time. This gives us one last definition:

Definition 3.3 (Computational Resilient Function)**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a *computational* (k, ε)*-RF* if and only if for every $L \in \left\{ {n \atop n-k} \right\}$ and $a \in \{0,1\}^{n-k}$ we have $f(\langle L^{a,n} \rangle) \cong^c_\varepsilon \langle U_m \rangle$.[4]

3.1.2 Exposure-Resilient Functions

All of the relaxations that we considered above had to do with the distinguishing step of the game that we set up. Let us consider instead relaxing the setup step by stipulating that instead of *setting* some of the bits of f's input and having the rest of the input chosen randomly, the entire input string r will be chosen randomly and A can *view* $n-k$ of r's bits. Functions that do well under this relaxation are called *exposure-resilient functions* (ERF's).

Once we make weaken A's ability in the setup step, the game becomes more flexible in two ways. First, the setup step could remain before the query step, but since the choice of r is independent of any action that A takes, the setup step could also come *after* the query step, producing a so-called "strong" ERF. Second, A may choose L, the set of bits that it is given permission to see, either all at once (that is, statically) or adaptively. We thus have four possible ways to lead up to the distinguishing step, each of which gives a different type of ERF. These four possibilities are described in Table 3.1.

[3] Recall that for two distributions $\langle A \rangle$ and $\langle B \rangle$, $\langle A \rangle \cong_\varepsilon \langle B \rangle$ means that $\Delta(\langle A \rangle, \langle B \rangle) \leq \varepsilon$.

[4] Recall that for two distributions $\langle A \rangle$ and $\langle B \rangle$, $\langle A \rangle \cong^c_\varepsilon \langle B \rangle$ means that no polynomial-time algorithm can distinguish $\langle A \rangle$ from $\langle B \rangle$ with advantage greater than ε.

	Weak	Strong
Static	A chooses L all at once, receives $[r]_L$, and then receives z	A receives z and then chooses L all at once and receives $[r]_L$
Adaptive	A chooses L adaptively while receiving the appropriate bits of r and then receives z	A receives z and then chooses L adaptively while receiving the appropriate bits of r

Table 3.1: Different Types of ERF's

Having put forth these four types of ERF's, we surely should ask whether each of them has perfect, statistical, and computational variants. Strictly speaking, of course, they all do in the sense that one can define each variant coherently. However, the real issue is whether we have really just defined twelve different types of ERF's along with our three RF's, and whether those ERF's and RF's might have anything to do with each other. This is the central question that will occupy us in the next chapter as we prove relationships among the various species in our zoo and exhibit constructions to differentiate between them. We will quickly discover that we are dealing with considerably fewer than fifteen different types of functions and that there is a great deal of structure governing their relationships, but first we must familiarize ourselves with the basics of the theory.

Let us write some definitions for these objects that talk about distributions rather than adversaries, as we did with resilient functions in the previous section. In the case of weakly static ERF's this task is straightforward:

Definition 3.4 (Weakly Static ERF[5])**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a *weakly static* (k,ε)*-ERF* if and only if for every $L \in \genfrac{\{}{\}}{0pt}{}{n}{n-k}$ we have:

$$([\langle U_n\rangle]_L, f(\langle U_n\rangle)) \approx ([\langle U_n\rangle]_L, \langle U_m\rangle)$$

In the strongly static case, we need a way to reflect the fact that A knows $f(r)$ when it chooses L. We can do this by creating a function $d\colon \{0,1\}^m \to \genfrac{\{}{\}}{0pt}{}{n}{n-k}$ that reflects A's decision-making process.

Definition 3.5 (Strongly Static ERF)**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a *strongly static* (k,ε)*-ERF* if and only if for every algorithm $d\colon \{0,1\}^m \to \genfrac{\{}{\}}{0pt}{}{n}{n-k}$ we have:

$$\big([\langle U_n\rangle]_{d(f(\langle U_n\rangle))}, f(\langle U_n\rangle)\big) \approx \big([\langle U_n\rangle]_{d(\langle U_m\rangle)}, \langle U_m\rangle\big)$$

Two small comments are in order here: first, the subroutine d is executed by the adversary and so it is computationally unbounded in the perfect and statistical settings, but must run in polynomial time in the computational setting. Second, we will treat the function d as non-randomized

[5] Note that the use of $\approx$ in this definition means that the statement ought to be one of equality in the perfect setting, statistical indistinguishability with parameter ε in the statistical setting, and computational indistinguishability with parameter ε in the computational setting. This caveat applies to the next three definitions as well.

because if some randomized d can distinguish between the two distributions then we can find a setting to which we can fix its coin flips so that it still distinguishes with the same advantage. These ideas also apply to the functions d that we employ in the following two definitions.

For the adaptive variants the definitions are a bit more involved:

Definition 3.6 (Weakly Adaptive ERF)**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a *weakly adaptive* (k,ε)*-ERF* if and only if for every algorithm $d\colon \{0,1\}^n \to \{0,1\}^*$ that can read at most $n-k$ of the bits of its input, we have:

$$(d(\langle U_n\rangle), f(\langle U_n\rangle)) \approx (d(\langle U_n\rangle), \langle U_m\rangle)$$

What we really imagine when we talk about d is an algorithm that simply chooses $n-k$ bits of its input adaptively and outputs them in the order that they appear in the input. However, we can allow d to do anything it wants for the following reason: Given d, let d' be an algorithm that makes all the same queries as d but outputs the results of those queries in the order it receives them and does no more. Then $d(\langle U_n\rangle)$ is a (deterministic) function of $d'(\langle U_n\rangle)$. Since we showed in Lemma 2.6 that applying a deterministic function to two distributions cannot make them any farther apart with respect to Δ, this shows that $(d'(\langle U_n\rangle), f(\langle U_n\rangle)) \approx (d'(\langle U_n\rangle), \langle U_m\rangle)$ implies $(d(\langle U_n\rangle), f(\langle U_n\rangle)) \approx (d(\langle U_n\rangle), \langle U_m\rangle)$ in the perfect and statistical settings. In the computational setting the conclusion also holds since any distinguisher that employs d could instead be said to employ d' and then do the work that d does on top of d' as part of its "distinguishing algorithm".

We define strongly adaptive ERF's by combining the two previous definitions:

Definition 3.7 (Strongly Adaptive ERF)**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a *strongly adaptive* (k,ε)*-ERF* if and only if for any algorithm $d\colon \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^*$ that can read at most $n-k$ of the bits of its first input, we have:

$$(d(\langle U_n\rangle, f(\langle U_n\rangle)), f(\langle U_n\rangle)) \approx (d(\langle U_n\rangle, \langle U_m\rangle), \langle U_m\rangle)$$

In this definition d is again given free latitude in what it does, even though we imagine an algorithm that simply uses its second argument to help it adaptively request $n-k$ bits of the first argument and then outputs those bits in the order in which it received them. It might initially seem here that d could output bits of its second argument and thereby cause the distribution on the left to be far from uniform because its two components are not independent; however, in this case both $(d(\langle U_n\rangle, f(\langle U_n\rangle)), f(\langle U_n\rangle))$ and $(d(\langle U_n\rangle, \langle U_m\rangle), \langle U_m\rangle)$ would be far from uniform (and close to each other) since d's second argument is $f(\langle U_n\rangle)$ in one case and $\langle U_m\rangle$ in the other.

The latter three definitions seem quite separate from the first in that to prove that some object satisfies them we would need to prove something about all algorithms of some sort. In the next section we will give an alternate characterization of statistical RF's and ERF's that, by relating them to randomness extractors, will enable us to solve this problem.

3.2 An Alternate Characterization

The statistical variants of both resilient and exposure-resilient functions are concerned with trying to produce a distribution that is statistically close to uniform using a random source that is somehow flawed. This is reminiscent of the goal of randomness extractors, and indeed it turns out that many of the objects we considered above can be characterized in the language of extractors. More generally, however, we find here that a key distinction between resilient functions and exposure-resilient functions is that we can characterize RF's in terms of worst-case behavior and ERF's in terms of average-case behavior. This will be crucial in the next chapter when we try to show some non-trivial relationships among these functions.

The alternate characterization presented here is proven only in the statistical setting, and some of it truly does not apply to the computational setting (though we will find out in Section 3.4 that this does not matter so much). What is true, however, is that we can use these all of these alternate definitions in the perfect setting by just setting $\varepsilon = 0$.

3.2.1 Resilient Functions: Worst-Case Extractors

In the case of statistical resilient functions the connection to randomness extractors is the easiest to see, for statistical RF's are simply a type of extractor, as the following proposition shows.

Proposition 3.8. *A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a statistical (k,ε)-RF if and only if it is a deterministic ε-extractor for (n,k) oblivious bit-fixing sources.*

Proof. Let $\langle S\rangle$ be an (n,k) oblivious bit-fixing source. Let $L \in \left\{ {n \atop n-k} \right\}$ be the set of the positions of the bits of $\langle S\rangle$ that are fixed, and let $a \in \{0,1\}^{n-k}$ be the value to which they are fixed. Then $\langle S\rangle = \langle L^{a,n}\rangle$ and so the definitions of statistical RF and deterministic extractor become identical. □

This can be seen as a worst-case guarantee on extraction in the sense that no matter which (n,k) oblivious bit-fixing source we choose, f will extract good randomness from it. This contrasts nicely with the ERF's which turn out give us *average*-case guarantees on the randomness they extract.

3.2.2 Weak ERF's: Average-Case Extractors

As in the previous section, the simplest ERF's to deal with are the weakly static statistical (k,ε)-ERF's, which turn out to be equivalent to average-case deterministic extractors for very simple classes of (n,k) oblivious bit-fixing sources.

Proposition 3.9. *A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a weakly static statistical (k,ε)-ERF if and only if for every $L \in \left\{{n \atop n-k}\right\}$, f satisfies:*

$$\mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k}\rangle} \left[\Delta\left(f(\langle L^{a,n}\rangle), \langle U_m\rangle\right)\right] \leq \varepsilon$$

Proof. As we showed in Lemma 2.6 in our preliminaries, given any two joint distributions $(\langle X\rangle, \langle Y\rangle)$, $(\langle X\rangle, \langle Y'\rangle)$, we can write

$$\Delta\left(\left(\langle X\rangle, \langle Y\rangle\right), \left(\langle X\rangle, \langle Y'\rangle\right)\right) = \mathop{\mathrm{E}}_{x \leftarrow \langle X\rangle} \left[\Delta\left(\langle Y\rangle|_{\langle X\rangle = x}, \langle Y'\rangle|_{\langle X\rangle = x}\right)\right]$$

If we set $\langle X\rangle = [\langle U_n\rangle]_L$, $\langle Y\rangle = f(\langle U_n\rangle)$, $\langle Y'\rangle = \langle U_m\rangle$, the statement reads

$$\Delta\left(\left([\langle U_n\rangle]_L, f(\langle U_n\rangle)\right), \left([\langle U_n\rangle]_L, \langle U_m\rangle\right)\right) = \mathop{\mathrm{E}}_{a \leftarrow [\langle U_n\rangle]_L} \left[\Delta\left(f(\langle U_n\rangle)|_{[\langle U_n\rangle]_L = a}, \langle U_m\rangle|_{[\langle U_n\rangle]_L = a}\right)\right]$$

Since $f(\langle U_n\rangle)|_{[\langle U_n\rangle]_L = a} = f(\langle L^{a,n}\rangle)$ and $\langle U_m\rangle$ and $\langle U_n\rangle$ are independent, this gives the desired result. □

Proposition 3.9 shows that a weakly static (k,ε)-ERF f is an average-case extractor for (n,k) oblivious bit-fixing sources in the sense that given any $L \in \left\{{n \atop n-k}\right\}$, we can create a collection of (n,k) oblivious bit-fixing sources $\mathcal{S}_L = \{\langle L^{a,n}\rangle : a \in \{0,1\}^{n-k}\}$ and we have the guarantee that f must extract randomness well "on average" over the uniform choice of a source $\langle S\rangle \in \mathcal{S}_L$.

In the weakly adaptive case, this idea holds as well, only the collections of sources about which we have average-case extraction guarantees are more complicated. To characterize these collections, we fist need to introduce a special class of functions called branching functions.

Definition 3.10 (Branching Function)**.** A function $\varphi\colon \{0,1\}^\ell \to \left\{{n \atop \ell}\right\}$ is a *branching function* if and only if, given any collection C of strings that all agree in their first i bits, φ satisfies the inequality $\left|\bigcap_{w \in C} \varphi(w)\right| \geq i+1$ for all $0 \leq i < \ell$.

Now, to state the average-case extractor version of the definition of weakly adaptive ERF's we can simply replace "every set L" with "every branching function φ".

Proposition 3.11. *A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a weakly adaptive statistical (k,ε)-ERF if and only if for every branching function $\varphi\colon \{0,1\}^{n-k} \to \left\{{n \atop n-k}\right\}$, f satisfies:*

$$\mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k}\rangle} \left[\Delta\left(f(\langle \varphi(a)^{a,n}\rangle), \langle U_m\rangle\right)\right] \leq \varepsilon$$

Proof. As we noted when we gave the definition of weakly adaptive ERF's, we may assume without loss of generality that the algorithm d from the definition simply selects $n-k$ bits adaptively and prints them out in the order that they appear in the input. Thus, by the same argument as in the previous proposition we can show that the definition of weakly adaptive statistical ERF is equivalent to:

$$\Delta\left((d(\langle U_n\rangle), f(\langle U_n\rangle)), (d(\langle U_n\rangle), \langle U_m\rangle)\right) = \mathop{\mathrm{E}}_{a\leftarrow\langle U_{n-k}\rangle}\left[\Delta\left(f(\langle U_n\rangle)|_{d(\langle U_n\rangle)=a}, \langle U_m\rangle\right)\right]$$

We have written $\langle U_{n-k}\rangle$ instead of $d(\langle U_n\rangle)$ under the expectation operator since the bits of $\langle U_n\rangle$ are independent and therefore $d(U_n)$ will equal $\langle U_{n-k}\rangle$ no matter which positions d chooses to read. We have also simplified $\langle U_m\rangle|_{d(\langle U_n\rangle)=a}$ to $\langle U_m\rangle$. Now let us deal with the two directions of the proposition separately:

($\Leftarrow$): Suppose we are given any algorithm d as in the definition of weakly adaptive ERF's. Then given any string $a \in \{0,1\}^{n-k}$, define $\varphi(a)$ to be the set of the positions of the bits that d requests when it is fed the bits of a one at a time in response to its queries. Since d adaptively requests positions, we know that if it receives the same bits in response to its queries then it will request the same positions. In other words, φ is a branching function. Since $\langle\varphi(a)^{a,n}\rangle = \langle U_n\rangle|_{d(\langle U_n\rangle)=a}$, we are done.

($\Rightarrow$): Suppose we are given any branching function φ as described above. We will build an algorithm $d\colon \{0,1\}^n \to \{0,1\}^*$ that adaptively requests $n-k$ positions of its input and prints them out with the property that $d(x) = a$ if and only if $[x]_{\varphi(a)} = a$. For any string w of length $0 \le |w| < n-k$, we can apply φ to all possible completions of w to a string of length $n-k$ with the guarantee that all the resulting sets will have at least $|w|+1$ elements in common. Let S_w denote the set of these elements. It is easy to see that for every $\beta \in \{0,1\}$ we have $S_w \supseteq S_{w\circ\beta}$.

We now can build d: we will have it initially request any element of S_e (where e is the string of length 0). At any other point in its execution, d will have received some string w of bits, and so we can have it request any element of S_w that has not yet been requested (which must be possible since $|S_w| \ge |w|+1$ and only $|w|$ bits have been requested so far). This d has the claimed properties and thus $f(\langle\varphi(a)^{a,n}\rangle) = f(\langle U_n\rangle)|_{d(\langle U_n\rangle)=a}$ and f is a weakly static statistical ERF as claimed. □

It is instructive to note that the forward direction of this proof does not work in the computational setting, because there is no guarantee that the algorithm d that we construct will be efficient. Thus, this characterization really is specific to the statistical setting. In Section 3.4 we will see, though, that the statistical setting is more central than the computational setting because we can easily construct computational ERF's from statistical ERF's.

We have seen how the weak ERF's are average-case extractors for different types of collections of sources. What, then, are the strong ERF's, and can we define them similarly?

3.2.3 The Strong ERF's

It may seem strange now, but our work in the next chapter (specifically, at the end of Section 4.2.2) will show us why the strong ERF's in fact cannot be characterized in terms of average-case deterministic extraction for (n, k) oblivious bit-fixing sources. However, we *can* finish our alternate characterization by defining the strong ERF's in terms of average-case behavior. This will prove quite useful later when we go about classifying our functions and will give us some insight into the differences between the strong and weak ERF's.

Let us first write our alternate definition of strongly static statistical ERF's.

Proposition 3.12. *A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is a strongly static statistical (k, ε)-ERF if and only if, for every function $d\colon \{0,1\}^m \to \left\{ {n \atop n-k} \right\}$, f satisfies:*

$$\frac{1}{2} \mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k} \rangle} \left[\sum_{z \in \{0,1\}^m} \left| \Pr[f(\langle d(z)^{a,n} \rangle) = z] - 2^{-m} \right| \right] \leq \varepsilon$$

Proof. The argument is a straightforward manipulation of the definition of statistical distance. We know that $\left([\langle U_n \rangle]_{d(f(\langle U_n \rangle))}, f(\langle U_n \rangle)\right) \cong_\varepsilon \left([\langle U_n \rangle]_{d(\langle U_m \rangle)}, \langle U_m \rangle\right) = \langle U_{n-k+1} \rangle$, so by appealing to the interpretation of statistical distance in terms of the L^1-norm (Proposition 2.5), we may write:

$$\begin{aligned}
\varepsilon &\geq \frac{1}{2} \sum_{\substack{a \in \{0,1\}^{n-k} \\ z \in \{0,1\}^m}} \left| \Pr\left[\left([\langle U_n \rangle]_{d(f(\langle U_n \rangle))}, f(\langle U_n \rangle) \right) = (a, z) \right] - 2^{-(n-k+m)} \right| \\
&= \frac{1}{2} \sum_{\substack{a \in \{0,1\}^{n-k} \\ z \in \{0,1\}^m}} \left| \Pr\left[f(\langle U_n \rangle) = z |_{[\langle U_n \rangle]_{d(z)} = a} \right] \cdot \Pr\left[[\langle U_n \rangle]_{d(z)} = a \right] - 2^{-(n-k+m)} \right| \\
&= \frac{1}{2} \sum_{\substack{a \in \{0,1\}^{n-k} \\ z \in \{0,1\}^m}} \left| \Pr\left[f(\langle d(z)^{a,n} \rangle) = z \right] \cdot 2^{-(n-k)} - 2^{-(n-k+m)} \right| \\
&= \frac{1}{2} \cdot \frac{1}{2^{n-k}} \sum_{a \in \{0,1\}^{n-k}} \sum_{z \in \{0,1\}^m} \left| \Pr\left[f(\langle d(z)^{a,n} \rangle) = z \right] - 2^{-m} \right| \\
&= \frac{1}{2} \mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k} \rangle} \left[\sum_{z \in \{0,1\}^m} \left| \Pr\left[f(\langle d(z)^{a,n} \rangle) = z \right] - 2^{-m} \right| \right]
\end{aligned}$$

□

Note that if the function d completely ignored the argument y (that is, if f were effectively a weakly static ERF) then this alternate definition reduces exactly to our alternate definition of weakly static ERF's, indicating that this really is the appropriate definition from our new viewpoint of average-case behavior.

Lastly, we mention that to turn this into a definition for strongly adaptive ERF's, we may simply replace $d(y)$ with $d(a, y)$ and stipulate that d be a branching function in its first argument. The proof is identical to that of Proposition 3.11.

Remark 3.13. We have seen that RF's are worst-case randomness extractors while weak ERF's are average-case extractors, and that strong ERF's are another sort of average-case object. A natural question to ask at this point is whether there might exist a worst-case analogue of the strong ERF's. There is something to this intuition, and we will return to it in detail in Section 4.2.3 after we sort out the world of RF's and ERF's.

3.3 Applications

Now that we have written down precise definitions for our functions, we can go into some of the applications of RF's and ERF's in detail.

3.3.1 Extraction of Randomness

If we look to the definitions given in our alternate characterization of the theory, we see that statistical RF's can extract randomness deterministically from oblivious bit-fixing sources and that weak statistical ERF's can extract randomness on average from various families of oblivious bit-fixing sources.

3.3.2 The Faulty Channel Problem

Suppose Alice and Bob need to share a random key $w \in \{0,1\}^m$ in order to carry out some cryptographic protocol, but the only way they can communicate privately is through a channel C that is somehow faulty. Suppose further that over some perfect but public channel Alice and Bob have shared an efficient description of a polynomial-time computable function $f\colon \{0,1\}^n \to \{0,1\}^m$ that is some sort of RF or ERF and that Alice chooses some $x \to \langle U_n \rangle$, sends x to Bob through C, and then both parties use $f(x)$ as the key for their protocol. Let us explore what kinds of faults in C can be overcome depending on the type of RF or ERF that f is. (In the following discussion we suppress the parameter ε, along with the obvious statements about the security being either perfect, statistical, or computational)

- If f is a weakly static (resp. adaptive) k-ERF, Alice and Bob will be able to thwart any interlocutor Eve who can statically (resp. adaptively) access up to $n - k$ bits of any message sent through C.

- If f is a k-RF, Alice and Bob will be able to thwart Eve even if she can *tamper* with up to $n - k$ bits of any message sent through C. They would also be able to preserve the integrity of their protocol if C is simply unreliable in the sense that it may incorrectly transmit up to $n-k$ bits of each message.

In all of these scenarios, the bits coming out of C to Bob are essentially a family of sources, which is why applying an RF or weak ERF to them gives us output that looks random. In the case of ERF's, this shows that ERF's with long outputs are basically extra-strength PRG's in the sense that wherever we would use a PRG G (for example, in a one time pad $E(x;r) = x \oplus G(r)$) we can replace G by a computational ERF f to make our protocol exposure resilient (the one-time pad would become $E(x;r) = x \oplus f(r)$).

3.3.3 Key Renewal Problem

Suppose that Alice and Bob already share a key, but they are worried that Eve has learned up to $n-k$ of its bits either adaptively or statically. If f is a weakly adaptive or static ERF, they can each separately apply f to their current key and obtain a new one that is secure once more. In fact, they can keep their key secure even if Eve is constantly learning bits of their key at some bounded rate: all they need to do is periodically apply f to the key with period slightly shorter than the amount of time it takes Eve to learn $n-k+1$ bits! Note that in order for the key not to shrink with every iteration, this problem requires $m = n$, and so f needs to be a computational ERF.

3.3.4 All-or-Nothing Transforms

The above constructions all only use RF's and weak ERF's. For what might we require strong ERF's? Well, in the above scenarios, our RF/ERF f provided us with exposure-resilient keys of all sorts, but they all had one element in common: the key required for the protocol was a uniformly random one. What about protocols that require keys with more structure than that? One solution is to treat whatever uniformly random bits are used to generate *those* keys as the "real" key and proceed as above. However, a more direct solution is to use a so-called *all-or-nothing transform* (AONT). An AONT is an efficiently invertible function with the property that if an adversary learns all but k bits of its output then he can determine nothing about the input. There are static, adaptive, perfect, statistical, and computational variants of this definition (there are not strong and weak variants since the adversary begins by receiving the output of the function in all cases). The formal definition of a static statistical AONT appears below.

Definition 3.14 (Static Statistical All-or-Nothing Transform)**.** A possibly randomized function $T \colon \{0,1\}^n \to \{0,1\}^m$ is a *k-AONT* if and only if

1. T is injective and efficiently invertible.
2. For any $L \in \left\{ {m \atop m-k} \right\}$ and any $x_0, x_1 \in \{0,1\}^n$ we have that

$$(x_0, x_1, [T(x_0)]_L) \cong_{\varepsilon} (x_0, x_1, [T(x_1)]_L)$$

In other words, $[T(x_0)]_L$ is indistinguishable from $[T(x_1)]_L$ for any x_0, x_1.

An important generalization of this is to give the adversary unrestricted access to some part of the output of the transform (the so-called "public" part) while restricting its access to the rest of the output (the "secret" part). Satisfyingly, Canetti et al. [9] show that if $f\colon \{0,1\}^n \to \{0,1\}^m$ is a weakly static statistical (k, ε)-ERF then the randomized function[6] $T(x;r) = (r, x \oplus f(r))$ is a static statistical AONT with secret part r and public part $x \oplus f(r)$. Further, Dodis, Sahai, and Smith [21] observe that the same construction suffices to build an adaptive statistical AONT as long as f is a *strongly* adaptive ERF.

3.3.5 Gap Secret Sharing

Suppose we require a random secret for some protocol and want to share it among n parties in a way such that if any $n-k$ or fewer of the parties collaborate they will gain no computational information about the secret, but if all the parties collaborate they will be able to calculate the secret. A scheme that achieves this goal is called a gap secret-sharing scheme (the "gap" is between $n-k$ and n), and we can of course build one by choosing $x \leftarrow \langle U_n \rangle$, distributing each bit of x to a different party, and using $f(x)$ as our secret (here f is a weak k-ERF).

This is nice, but once again it only works if our secret is drawn from the uniform distribution. If we use an AONT we can achieve the same goal with an arbitrary secret, for if s is our secret and T is an AONT, we can distribute the bits of $T(x)$ to the parties.

AONT's have wide-ranging applications that are beyond the scope of this work. For an exposition on the topic, see [22].

3.4 Constructions

Here we discuss how to actually build polynomial-time computable instantiations of the objects we have defined and learn a bit more about them in doing so. We will first build a perfect ERF and discover that in the case of linear functions on $\mathbb{F}_2^n$, ERF's are the same as error-correcting codes. Next, by relaxing our requirements to statistical indistinguishability, we will be able to build ERF's with much better parameters. Finally, we will see that in the computational setting we can use a pseudorandom generator to build an ERF with ideal parameters and that in fact the existence of computational RF's and ERF's is equivalent to the existence of PRG's.

Throughout the section, we construct mostly ERF's for the sake of consistency in comparisons of parameters. However, we prove the correctness of some of our constructions by establishing that

[6] Recall that $T(x;r)$ indicates that T is a function of x that uses the uniformly random bits r.

they are RF's. This is justified in the following chapter, in which we show that every perfect (resp. statistical) RF is also a perfect (resp. statistical) ERF (see Theorems 4.1 and 4.7).[7]

Recall the that there are two ways that we wish to optimize the parameters of an ERF $f\colon \{0,1\}^n \to \{0,1\}^m$ First of all, we want the number of protected bits k to be as small as possible relative to n. Second, we want the output size m to be as large as possible. With these goals in mind, let us see what we can achieve in each setting...

3.4.1 A Perfect ERF

Despite the title of this section, we are going to construct our perfect ERF by constructing a perfect RF (which is therefore a perfect ERF by a trivial reduction). This is no accident: in the next chapter we will show that all the different types of perfect ERF's are in fact equivalent to perfect RF's. For the time being, however, we will have to be content with a construction that seems stronger than what we need.

What sorts of parameters can we expect from a perfect k-RF $f\colon \{0,1\}^n \to \{0,1\}^m$? As we mentioned earlier in this chapter, when $m = 1$ and $k \geq 1$ then $f(w) = \bigoplus_i [w]_i$ is a perfect k-RF. Additionally, if we want $k = n - 1$ and $m \leq k$, we can let f return the exclusive-or of any m consecutive pairs of bits of its input. These examples suggest that $m \leq k$ always, and a simple argument confirms this: if $f(\langle L^{a,n}\rangle)$ is to equal $\langle U_m\rangle$, then the support of $\langle L^{a,n}\rangle$ must be of size at least 2^m. (We will see that the same bound applies in the statistical setting for $\varepsilon < 1/2$.) This is a serious limitation as far as we're concerned, because ideally we'd like our ERF's to give us as many "random" bits as possible.

Towards a Construction

A slightly different way to think about a perfect RF is as a function that, when restricted to any set $L^{a,n}$, is surjective and maps the same number of strings to each element in its target space (we call this property *regularity*). The following simple lemma shows that if we restrict our search to the linear functions over $\mathbb{F}_2^n$, we are essentially looking for an m-by-n matrix with the property that any k of its columns span $\mathbb{F}_2^m$.

Lemma 3.15. *Let $M\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a linear function represented by an $m \times n$ matrix whose columns span $\mathbb{F}_2^m$. Then M is 2^{n-m}-regular (i.e. M maps exactly 2^{n-m} elements of $\mathbb{F}_2^n$ to each element of $\mathbb{F}_2^m$).*

Proof. $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$ are of course groups under addition, and M is a homomorphism between them. Since $\dim(\operatorname{Im} M) = m$, we know that $\dim(\ker M) = n - m$. Let $K = \ker M$. Since K is a subgroup of $\mathbb{F}_2^n$, all of its cosets are the same size (namely, $|K| = 2^{n-m}$). Each of these corresponds to the pre-image under M of a different $x \in \mathbb{F}_2^m$, so we are done. □

[7] The reduction is actually quite simple and appears in a different chapter only for structural reasons.

Fortunately, the theory of error-correcting codes gives us such functions, and within the world of linear functions they in fact turn out to be *equivalent* to perfect RF's. (For background information on error-correcting codes, including their definition and some of their basic properties, see Appendix A.)

Theorem 3.16. *Let M be an $m \times n$ matrix. Define a function $f\colon \{0,1\}^n \to \{0,1\}^m$ by $f(r) = M \cdot r$. Then f is a perfect k-RF if and only if the map $N\colon \{0,1\}^m \to \{0,1\}^n$ sending $x \in \{0,1\}^m$ to $x^T \cdot M \in \{0,1\}^n$ is an error-correcting code with minimum distance $d \geq n-k+1$.*

Proof. ($\Leftarrow$): Since $N(0^m) = 0^n$, any non-trivial linear combination of the rows of M must have at least d ones. Thus, we can remove any $d-1 \geq n-k$ of the columns of M, call the new matrix M', and know that its rows are linearly independent since any non-trivial linear combination of them will have at least one entry equal to 1. Since the m rows of M' are linearly independent, its row rank (and therefore its column rank) is $m = \dim \mathbb{F}_2^m$, allowing us to apply Lemma 3.15 to find that the map $x \mapsto M' \cdot x$ is regular . This shows that f is regular on $L^{a,n}$ for any $L \subset [n]$ of size $d-1 \geq n-k$ and $a = 0^{d-1}$. But in fact the result holds for any a, for we may assume without loss of generality that $L = [d-1]$ and then write $L^{a,n} = a \circ 0^{n-d+1} + L^{0^{d-1},n}$. Then by the linearity of f we have that $f(L^{a,n}) = M \cdot (a \circ 0^{n-d+1}) + f(L^{0^{d-1},n})$ which is also regular since it differs from a regular map by addition of a constant, which is a permutation of $\mathbb{F}_2^m$.

($\Rightarrow$): Suppose for contradiction that we have $v \in \{0,1\}^m$ non-zero and with the property that $v^T \cdot M$ has fewer than $n-k+1$ ones. Then we can find a set of $n-k$ columns to remove such that the remaining matrix M' has a linear dependency among its rows. So the row rank (and therefore the column rank) of M' is strictly less than m, implying that $r \mapsto M' \cdot r$ is not surjective. This means that if we let L be the set of indices of the columns we removed from M, we have that $f(L^{0^{n-k},n})$ is not surjective, implying that it is not regular since 0^m is in its image but there exists some string in $\{0,1\}^m$ that is not. Therefore, f is not a perfect k-RF. □

Given this result, it is satisfying to note that the "Singleton bound" on error-correcting codes (Proposition A.2), which states that $m \leq n-d+1$, gives us the previously discussed bound $m \leq k$ for perfect k-RF's, and vice versa.

What kinds of parameters does the above theorem give us? Proposition A.3 in Appendix A shows that for an error-correcting code corresponding to a linear function $f\colon \{0,1\}^n \to \{0,1\}^m$ we have that $m > \log n + 1$ implies $d < \lceil n/2 \rceil$, which implies that in terms of f's parameters we are limited to $k > \lceil n/2 \rceil$ as long as we insist on having any appreciable sort of output length.

3.4.2 Two Weakly Static Statistical ERF's

In our perfect ERF construction, we had two serious limitations: $k \geq n/2$, and $m \leq k$. This is (mostly) not a deficiency of the constructions themselves, for $m \leq k$ follows necessarily from the definition, and Chor et al. [23] show that even for $m = 2$ we must have $k \geq n/3$; that is, one third of the bits of the input must remain secret in order to get even *two* random bits.

Relaxing our definition of indistinguishability to the statistical one allows us to overcome these two limitations, but there is no known construction that addresses both at once. We will therefore present two constructions:

- Our first construction is a weakly static statistical ERF with output length γk for any constant $0 < \gamma < 1$ and exponentially small ε that works when $k \in \omega(\log n)$. This output length is essentially optimal for the statistical setting, for if we have $m > k$, then the support of $f(\langle L^{a,n} \rangle)$ is less than half of $\{0,1\}^m$ and so $\Delta(f(\langle L^{a,n}\rangle), \langle U_m \rangle) \geq 1/2$ for all L and a, which means that when we average over a in Proposition 3.9, we will get $\varepsilon \geq 1/2$.
- Our second construction is a statistical RF that works for any k (including even *constant* k), but pays a price in output length, outputting only $(\log k)/4$ random bits. Recall that this function will, by virtue of being a statistical RF, also be a weakly static statistical ERF as desired.

The First Construction: Almost Optimal Output Length

Our first construction takes a strong extractor for sources with min-entropy k and uses it directly as a weakly static statistical ERF.[8] It makes sense that such an object would help us obtain an ERF: part of its input has exposure-resilient properties (that is, seeing the seed of a strong extractor does not help any adversary as long as that seed was chosen randomly and independently of the rest of the input), and the rest of its input has resilient properties (setting some of the non-seed input bits of a strong extractor for k-sources does not cause the output to deviate significantly from uniform).

Before we prove this though, sketching it in more detail will be helpful: what we do is use part of f's input (say the first s bits) to seed the extractor. The above discussion shows that if this part of the input is seen by the adversary then we lose nothing. As for the rest of the input: regardless of which bits have been seen by the adversary, there will be at least $k - s$ bits that are still hidden, and so, conditioned on the values of the revealed bits, we are left with a $(k - s)$-source from which we can extract provided our extractor is good enough. It is important to note two things here: first, that the construction is not a resilient function because the strong extractor property, while it protects against exposure of the seed, relies on the seed being chosen uniformly at random; and second, that we must have s be small enough so that our $(k - s)$-source has enough entropy left in it from which to extract. This argument is formalized in the following theorem.

Theorem 3.17. *Let $h\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ be any strong ε-extractor for ℓ-sources. Then the function $f\colon \{0,1\}^{n+s} \to \{0,1\}^m$ that is defined by setting $f(w) = h([w]_{[s]}, [w]_{[n+s]\setminus[s]})$ is a weakly static statistical (k, ε)-ERF for $k \geq \ell + s$.*

Proof. Fix any $L \in \left\{ {n+s \atop n+s-k} \right\}$. We will show that f is an average-case extractor for the set of sources $\{\langle L^{a,n+s}\rangle : a \in \{0,1\}^{n+s-k}\}$, which will satisfy our alternate characterization for weakly static statistical ERF's.

[8] We cover min-entropy in Section 2.3.1.

Let $L_1 = L \cap [s]$ and let $L_2 = L - L_1$. For any $a \in \{0,1\}^{n+s-k}$, let a_i denote $[L^{a,n+s}]_{L_i}$.[9] We have that $|L_2| \le |L| = n + s - k$. Therefore, the number of un-fixed bits in $\langle L_2^{a_2,n} \rangle$ is at least $n - |L_2| = k - s \ge \ell$, so $\langle L_2^{a_2,n} \rangle$ is an ℓ-source for any $a_2 \in \{0,1\}^{|L_2|}$.

Since h is a strong extractor for such sources, we have by definition that for any $y \in \{0,1\}^{|L_2|}$:

$$\begin{aligned} & (\langle U_s \rangle, h(\langle U_s \rangle, \langle L_2^{y,n} \rangle)) && \cong_\varepsilon (\langle U_s \rangle, \langle U_m \rangle) \\ \Rightarrow\ & ([\langle U_s \rangle]_{L_1}, h(\langle U_s \rangle, \langle L_2^{y,n} \rangle)) && \cong_\varepsilon ([\langle U_s \rangle]_{L_1}, \langle U_m \rangle) \end{aligned}$$

And rewriting with the help of Lemma 2.6 from our preliminaries, we arrive at

$$\mathop{\mathrm{E}}_{x \leftarrow \langle U_{|L_1|} \rangle} \left[\Delta \left(h \left(\langle L_1^{x,s} \rangle, \langle L_2^{y,n} \rangle \right), \langle U_m \rangle \right) \right] \le \varepsilon$$

If this inequality holds for each $y \in \{0,1\}^{|L_2|}$ then surely it holds when we take expectation over y, so we may write:

$$\begin{aligned} \varepsilon \ &\ge \mathop{\mathrm{E}}_{y \leftarrow \langle U_{|L_2|} \rangle} \left[\mathop{\mathrm{E}}_{x \leftarrow \langle U_{|L_1|} \rangle} \left[\Delta \left(h \left(\langle L_1^{x,s} \rangle, \langle L_2^{y,n} \rangle \right), \langle U_m \rangle \right) \right] \right] \\ &= \mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k} \rangle} \left[\Delta \left(h \left(\langle L_1^{a_1,s} \rangle, \langle L_2^{a_2,n} \rangle \right), \langle U_m \rangle \right) \right] \\ &= \mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k} \rangle} \left[\Delta \left(f \left(\langle L^{a,n+s} \rangle \right), \langle U_m \rangle \right) \right] \end{aligned}$$

And so by Proposition 3.9 f is a weakly static statistical (k, ε)-ERF, as desired. □

Theorem 3.17 allows us to establish the existence of weakly static statistical ERF's with very good parameters provided we have good strong extractors. The parameters we seek are achieved by the strong extractor construction of Guruswami, Umans, and Vadhan [17] stated in Theorem 2.21, which obtains output size $m \ge (1-\alpha)\ell$ for any constant α while keeping the seed length $s \le \log n + O(\log(\ell/\varepsilon))$. Armed with this strong extractor, let us give an explicit construction of a weakly static statistical ERF.

Theorem 3.18. *For every constant $0 < \gamma < 1$, and for all n and k satisfying $k \le n$ and $k \in \omega(\log n)$, there exists a polynomial-time computable weakly static statistical (k, ε)-ERF $f\colon \{0,1\}^n \to \{0,1\}^m$ with $m \ge \gamma k$ and $\varepsilon \in 2^{-\Omega(k)}$ for sufficiently large n.*

Proof. We can invoke Theorem 2.21 with $\ell = k$ and $\varepsilon = 2^{-\alpha k}$ to get a strong ε-extractor for k-sources $h\colon \{0,1\}^s \times \{0,1\}^{n-s} \to \{0,1\}^m$ with $m \ge \gamma k$ and with $s \le \log n + O(\log k + \alpha k)$. We know then that there is a constant $M > 0$ such that, for sufficiently large n, $s \le \log n + M(\log k + \alpha k)$. Setting $\alpha = 1/(cM)$ for a constant c to be determined later and using that $k \in \omega(\log n)$, we obtain that asymptotically $s \le 2k/c$.

[9] Recall here that $[L^{a,n+s}]_{L_i}$ has only one element since $L_i \subset L$, and so the notation refers not to the set containing that element but to the element itself. That is, $[L^{a,n+s}]_{L_i}$ is the substring of a that corresponds to bits with positions in L_i.

We now apply Theorem 3.17 to find that h is a weakly static statistical $((1+\frac{2}{c})k, 2^{-\Omega(k)})$-ERF from n bits to γk bits for sufficiently large n. We can reparametrize and replace k with $k/(1+\frac{2}{c})$ to see that h is a weakly static $(k, 2^{-\Omega(k)})$-ERF with output length $k\gamma/(1+\frac{2}{c})$. Choosing γ sufficiently close to 1 and c sufficiently large, we can make the constant factor in the output size arbitrarily close to 1, as desired. □

The construction of Theorem 3.18 is quite nice: it allows us to get the output size of f to be any fixed percentage of k that we want, even while keeping ε exponentially small in $\Omega(k)$. Is this the best we can do in terms of output length using strong extractors? The following result of Radhakrishnan and Ta-Shma [24] will help us show that indeed it is.

Theorem 3.19 (Radhakrishnan and Ta-Shma [24])**.** *Let $E\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ be a seeded ε-extractor for ℓ-sources. Then $\ell + s - m \geq 2\log(1/\varepsilon) - O(1)$.*

This bound makes some sense—it tells us that any seeded extractor (including any strong extractor) *must* have some entropy loss, and that that entropy loss is inversely related to ε. It means, though, that if we construct a weakly static statistical ERF from a strong extractor for ℓ-sources with seed length s using Theorem 3.17, there is a constant $C > 0$ such that we will never be able to obtain $k-m < 2\log(1/\varepsilon)+C$ since the ERF that comes out of the theorem has $k \geq \ell+s$. This means that if we want to keep $\varepsilon = 2^{-\Omega(k)}$ then we are limited to $k - m \geq \Omega(k)$. In other words, the explicit construction given in Theorem 3.18 is in some sense the best result that we can squeeze out of the reduction of Theorem 3.17 in terms of output length. However, as we have discussed, this construction does not work for all k. Let us turn now to our second construction, which trades output length in exchange for requiring less entropy in its input.

The Second Construction: Protecting Fewer Input Bits

Our second construction is the deterministic extractor for (n, k) oblivious bit-fixing sources presented in Theorem 2.18, which by Proposition 3.8 is a statistical RF. We re-state Theorem 2.18 here and then present a sketch of its proof.

Theorem 3.20 (Restatement of Theorem 2.18)**.** *For every k and n satisfying $n \geq k > 0$, there exists a polynomial-time computable deterministic ε-extractor for (n, k) oblivious bit-fixing sources $E\colon \{0,1\}^n \to \{0,1\}^m$ with $m \geq \log k/4$ and $\varepsilon \leq 2^{-\sqrt{k}}$.*

Sketch of Proof. The extractor is actually quite simple: all it does is use the input to take a walk on a cycle of length $\ell = d(k^{1/4})$, where $d(x)$ is the smallest odd number that is greater than or equal to x. It is easy to see that the positions of the fixed bits do not affect the random walk (since we can take the steps in any order); once we see this we can assume that the fixed bits are all at the beginning of the input, which shows us that their values also do not matter since they'll only affect our starting point, which is arbitrary anyway. Therefore we can, without loss of generality, ignore the fixed bits. What we are left with is the task of bounding the distance from uniform of the distribution on the vertices of the cycle that arises if we start at an arbitrary vertex and take k random steps. This explains why we chose the cycle length to be odd: random walks on even-length cycles do not converge to uniform because with every step they alternate between even-labeled vertices and odd-labeled vertices.

Bounding the speed of convergence of our random walk to the uniform distribution on the vertices is a matter of linear algebra: first we can show that if we have a distribution on the vertices that is δ away from uniform and then take a random step, the resulting distribution is at most $\lambda(P)\delta$ away from uniform (in terms of L^2 distance between distributions) where $\lambda(P)$ is the second-largest eigenvalue of the transition matrix P of the cycle. This allows us to bound the L^2 distance from uniform after k steps by $\lambda(P)^k$. Next, we bound $\lambda(P)$ by $\cos(\pi/\ell)$ (cf. [25]) and show that $\cos(\pi/\ell) \leq \exp(-\frac{\pi^2}{2\ell^2})$ (cf. [26]).

Finally, we show that bounding the L^2-norm of the difference between two distributions by some A implies that their statistical distance is bounded by $\frac{1}{2}A\sqrt{\ell}$. We can then combine this with our bound of $\lambda(P)^k \leq \exp(-\frac{\pi^2 k}{2\ell^2})$ to obtain that $\varepsilon \leq \frac{1}{2}k^{\frac{1}{8}} \exp\left(-\frac{\pi^2\sqrt{k}}{2}\right)$, which by some algebra and bounds on e^x can be shown to be at most $2^{-\sqrt{k}}$. □

Can we improve on the parameters of these two constructions by constructing a weakly static statistical ERF that both works for all $0 < k \leq n$ *and* has output size proportional to k? Addressing this question will be our main goal in Chapter 5, where we will explore in detail whether we can take an existing (k, ε)-ERF f and make k any smaller while preserving the exposure-resilience properties of f.

3.4.3 Obtaining Computational RF's and ERF's

In the previous section we managed to construct an ERF with some nice parameters. In particular, the number of "protected" bits k can be almost logarithmic in n and the output length is linearly related to k. However, we wish to construct ERF's with *arbitrary* output lengths! The way we do this of course is to relax our definition of indistinguishability one more time and resort to pseudorandom generators (PRG's). We have already done the hard work of generating a distribution that is almost uniform. Now all we need to do is apply a PRG to that distribution to get a pseudorandom string of the length that we want. As in the case of linear resilient functions, we can close the loop by proving that the existence of (k, ε)-ERF's with output length greater than k implies the existence of PRG's. (For the definition of PRG's, see Section 2.4.)

In the statement of the following proposition we introduce a system of acronyms for referring to the different types of ERF's: WSERF means weakly static ERF, WAERF means weakly adaptive ERF, and so on. We will occasionally use these acronyms for brevity's sake in the remainder of this work as well.

Proposition 3.21. *Let $f\colon \{0,1\}^n \to \{0,1\}^m$ be a statistical (k,ε)-RF (resp. WSERF, WAERF, SSERF, SAERF) and let $G\colon \{0,1\}^m \to \{0,1\}^\ell$ be an ε' pseudorandom generator. Then the function $G \circ f$ is a computational $(k, \varepsilon + \varepsilon')$-RF (resp. WSERF, WAERF, SSERF, SAERF).*

Proof. In the case that f is a statistical RF the proof is straightforward: for all $L \in \left\{ {n \atop n-k} \right\}$ and $a \in \{0,1\}^{n-k}$ we know that $f(\langle L^{a,n}\rangle) \cong_\varepsilon \langle U_m\rangle$, so since G is deterministic, we have that $G(f(\langle L^{a,n}\rangle)) \cong_\varepsilon G(\langle U_m\rangle)$ (see Proposition 2.6). Therefore, if some efficient D, together with some $L \in \left\{ {n \atop n-k} \right\}$, has advantage greater than $\varepsilon + \varepsilon'$ in distinguishing $G(f(\langle L^{a,n}\rangle))$ from $\langle U_\ell\rangle$, then it has advantage greater than ε' in distinguishing $G(\langle U_m\rangle)$ from $\langle U_\ell\rangle$, contradicting that G is an ε'-PRG.

The proof for the ERF's is similar but requires a few additional steps. We first assume that f is an SAERF, and let $d\colon \{0,1\}^n \times \{0,1\}^m \to \left\{ {n \atop n-k} \right\}$ be any polynomial-time algorithm that reads no more than $n-k$ of the bits of its first input parameter. We can modify d so that before doing anything, it applies G, which is efficient and deterministic, to its second input parameter. The function f, being an SAERF, is still resistant to this new function, so we may write

$$\left([\langle U_n\rangle]_{d(\langle U_n\rangle, G(f(\langle U_n\rangle)))}, f(\langle U_n\rangle)\right) \cong_\varepsilon \left([\langle U_n\rangle]_{d(\langle U_n\rangle, G(\langle U_m\rangle))}, \langle U_m\rangle\right)$$

Now consider now the following three distributions:

$$\begin{aligned}
\langle A_d\rangle &= \left([\langle U_n\rangle]_{d(\langle U_n\rangle, G(f(\langle U_n\rangle)))}, G(f(\langle U_n\rangle))\right)\\
\langle B_d\rangle &= \left([\langle U_n\rangle]_{d(\langle U_n\rangle, G(\langle U_m\rangle))}, G(\langle U_m\rangle)\right)\\
\langle C_d\rangle &= \left([\langle U_n\rangle]_{d(\langle U_n\rangle, \langle U_\ell\rangle)}, \langle U_\ell\rangle\right)
\end{aligned}$$

Since G is deterministic, $\langle A_d\rangle$ is a deterministic function of $\left([\langle U_n\rangle]_{d(\langle U_n\rangle, G(f(\langle U_n\rangle)))}, f(\langle U_n\rangle)\right)$ and $\langle B_d\rangle$ is a deterministic function of $\left([\langle U_n\rangle]_{d(\langle U_n\rangle, G(\langle U_m\rangle))}, \langle U_m\rangle\right)$, so we know that $\langle A_d\rangle \cong_\varepsilon \langle B_d\rangle$ by the exposure-resilient property of f (along with Proposition 2.6). Now suppose for contradiction that $G \circ f$ is not a computational SAERF; that is, that some efficient algorithm D with an appropriate efficient subroutine $s\colon \{0,1\}^n \times \{0,1\}^m \to \left\{ {n \atop n-k} \right\}$ that reads only $n-k$ bits of its first argument could distinguish between $\langle A_s\rangle$ and $\langle C_s\rangle$ with advantage greater than $\varepsilon + \varepsilon'$. From this we see that D must be able to distinguish between $\langle B_s\rangle$ and $\langle C_s\rangle$ with advantage greater than ε'.

Since $\langle U_n\rangle$ is independent of both $\langle U_m\rangle$ and $\langle U_\ell\rangle$, the first component of both $\langle B_s\rangle$ and $\langle C_s\rangle$ can be replaced with $\langle U_{n-k}\rangle$. This is because regardless of which set s chooses, $\langle U_n\rangle$ restricted to that set consist of $n-k$ uniformly random bits that are independent of $\langle U_m\rangle$ and $\langle U_\ell\rangle$. This gives us that D distinguishes between $(\langle U_{n-k}\rangle, G(\langle U_m\rangle))$ and $(\langle U_{n-k}\rangle, \langle U_\ell\rangle)$ with advantage greater than ε' and so to distinguish whether a given string z came from $G(\langle U_m\rangle)$ or $\langle U_\ell\rangle$ with that advantage we can simply choose $r \leftarrow \langle U_{n-k}\rangle$ and return $D(r,z)$. We have now contradicted the security of G as an ε'-PRG.

To obtain this result for the other types of ERF's, we can simply repeat this argument with further restrictions on the functions d and s. For example, in the case of SSERF's, we can stipulate that d must ignore its first parameter altogether. Thus, we have shown the result in its full generality. □

Thanks to this proposition, we can, using existing PRG constructions (see Theorem 2.25), upgrade one of our "pet" ERF's—the weakly static statistical ERF of Theorem 3.18—to a polynomial-time computable weakly static computational (k, ε)-ERF mapping $\{0,1\}^n$ to $\{0,1\}^m$ for any k, n, and m, provided that $k \in \omega(\log n)$, $k \leq n$, $n \in k^{O(1)}$ and $m \in n^{O(1)}$.

Before we prove the full equivalence between PRG's, RF's, and ERF's, we first cite a result of Goldreich [27].

Theorem 3.22 (Goldreich[27])**.** *The following two conditions are equivalent:*

1. *The existence, for all $n \in \mathbb{N}$, of an ε-pseudorandom generator from n to $\ell(n) > n$ bits with ε negligible in n.*[10]
2. *The existence of a pair of polynomial-time constructible ensembles of distributions $\langle X_n \rangle$ and $\langle Y_n \rangle$ such that $\langle X_n \rangle \cong^c_\varepsilon \langle Y_n \rangle$ for ε negligible in n but, for some constant $c > 0$, it holds that $\Delta(\langle X_n \rangle, \langle Y_n \rangle) \geq 1/n^c$ for sufficiently large n.*[11]

We are now ready to show that computational RF's and ERF's are equivalent to PRG's from the point of view of complexity assumptions. This makes sense since a computational ERF with $k = n$ (that is, when all the input bits are chosen at random and protected) and output length greater than n is a PRG by definition. What we do here is formalize the idea that since the ERF works even when some of the bits are revealed, it is strong enough to give a PRG even though its output may not be longer than its input.

Theorem 3.23. *The following are equivalent:*

1. *There exist functions $k(n) \leq n$, $\ell(n) > k(n)$, and $\varepsilon(n)$ negligible in n such that, for all $n \in \mathbb{N}$, there exists a polynomial-time computable function from n bits to $\ell(n)$ bits that is a computational:*
   (a) *(k, ε)-RF.*
   (b) *Strongly adaptive (k, ε)-ERF.*
   (c) *Strongly static (k, ε)-ERF.*
   (d) *Weakly adaptive (k, ε)-ERF.*
   (e) *Weakly static (k, ε)-ERF.*
2. *There exists a function $\ell(n) > n$ such that, for all $n \in \mathbb{N}$, there exists a polynomial-time computable ε-pseudorandom generator from n bits to $\ell(n)$ bits with ε negligible in n.*

[10] Recall that ε is negligible in n if and only if for every constant $c > 0$ we have $\varepsilon(n) < 1/n^c$ for sufficiently large n.

[11] See Section 2.2.3 for a definition of polynomial-time constructible ensembles and a discussion of $\approx^c_\varepsilon$ and computational indistinguishability.

Proof. Proposition 3.21, together with the existence of the polynomial-time computable deterministic extractors presented in the Preliminaries for (n, k) oblivious bit-fixing sources (i.e. statistical RF's) that have error exponential in n, shows that $(2) \Rightarrow (1a)$ and $(2) \Rightarrow (1b)$. By trivial reductions, $(1b) \Rightarrow (1c) \Rightarrow (1e)$ and $(1b) \Rightarrow (1d) \Rightarrow (1e)$, so we need only show that $(1e) \Rightarrow (2)$.

$(1e) \Rightarrow (2)$: The hypothesis implies the existence, for any $L \in \left\{ {n \atop n-k} \right\}$, of the pair of polynomial-time constructible ensembles $\{([\langle U_n \rangle]_L, f(\langle U_n \rangle)) : n \in \mathbb{N}\}$ and $\{([\langle U_n \rangle]_L, \langle U_\ell \rangle) : n \in \mathbb{N}\}$ with $([\langle U_n \rangle]_L, f(\langle U_n \rangle)) \cong^c_\varepsilon ([\langle U_n \rangle]_L, \langle U_\ell \rangle)$ for ε negligible in n. However, since all the entropy in the former distribution comes from $\langle U_n \rangle$, the size of its support is at most 2^n. In contrast, the number of bits of entropy in the latter distribution is at least $n - k + \ell > n$, so the support of the former is at most half of the support of the latter, implying that the statistical distance between the two distributions is always at least 1/2. Since this is a constant, it is more than enough to apply Theorem 3.22 and deduce the existence of PRG's. □

3.5 References

Perfect RF's were introduced independently by Chor et al. [23] and Bennett, Brassard, and Robert [28]. Weakly static ERF's were introduced by Canetti et al. [9], though they were referred to simply as ERF's. Dodis, Sahai, and Smith [21], extended RF's to the statistical setting and introduced in the same paper [21] the notion of weakly/strongly adaptive/static statistical ERF's in terms of fooling adversaries. The definitions we give of the strongly static, strongly adaptive, and weakly adaptive ERF's that state those notions in terms of distributions, however, are original. The alternate characterization that we present is also original, as is the notion of branching functions. The secret-sharing scheme application of AONT's was pointed out in [9] by Canetti et al., who also sketch the ⇐ direction of Theorem 3.16. Both directions of the theorem are proven by Chor et al. in [23] using the Vazirani XOR-Lemma. However, our proof of the ⇒ direction, which does not use the Vazirani XOR-Lemma, is original. AONT's were introduced by Rivest [29]. Theorem 3.17 was proven in 2001 by Canetti et al. [9], but the proof presented here, which uses our alternate characterization, is original. The construction of a weakly static computational ERF using a PRG is presented in the same paper of Canetti et al. [9]; the extension of the idea to RF's is a well-known fact about extractors, but its extension to the other types of computational ERF's is original. The same is true of Theorem 3.23.

Chapter 4

Relationships Among Resilient and Exposure-Resilient Functions

Now that we have some intuition for what RF's and ERF's are and how they work, it is time to understand how the different notions of resilience and exposure-resilience relate to one another. We show in this chapter that in the setting of perfect security all of the notions introduced are completely equivalent. We then go on to address the statistical case in which, while each notion is distinct, there are very clear relationships among the different variants. The alternate definitions developed in Section 3.2 will be instrumental in proving the relationships we posit.

4.1 In a Perfect World...

In the world of perfect ERF's, things work out as neatly as we could possibly hope: all five types of functions are the same! This is the reason that it did not matter whether we constructed a perfect RF or a perfect ERF in Section 3.4.1.

Theorem 4.1. *For every function $f\colon \{0,1\}^n \to \{0,1\}^m$, the following are equivalent:*

1. *f is a perfect k-RF*
2. *f is a strongly adaptive perfect k-ERF*
3. *f is a weakly adaptive perfect k-ERF*
4. *f is a strongly static perfect k-ERF*
5. *f is a weakly static perfect k-ERF*

Proof. By trivial reductions we have (2) ⇒ (3) ⇒ (5) as well as (2) ⇒ (4) ⇒ (5). We will prove the theorem by showing (5) ⇒ (1) and (1) ⇒ (2).

(5) ⇒ (1): Let us treat f as a weakly static *statistical* ERF but set $\varepsilon = 0$ and apply our alternate definition (Proposition 3.9). This gives that for all $L \in \left\{ {n \atop n-k} \right\}$,

$$\mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k} \rangle} [\Delta(f(\langle L^{a,n} \rangle), \langle U_m \rangle)] = 0$$

Of course, since the expression inside the expectation operator is always non-negative, this implies that $\Delta(f(\langle L^{a,n}\rangle), \langle U_m \rangle) = 0$ for all L and a, which is what we desire.

(1) ⇒ (2): Similarly, we know by the alternate definition of strongly adaptive statistical ERF's (Section 3.2.3) with $\varepsilon = 0$ that it suffices to show that f satisfies the following equation for every function $d\colon \{0,1\}^{n-k} \times \{0,1\}^m \to \left\{ {n \atop n-k} \right\}$ that is a branching function in its first parameter:

$$\mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k} \rangle} \left[\sum_{y \in \{0,1\}^m} \left| \Pr[f(\langle d(a,y)^{a,n} \rangle) = y] - 2^{-m} \right| \right] = 0$$

But since f is a perfect k-RF and $|d(a,y)| = n - k$, we have that $\Pr[f(\langle d(a,y)^{a,n}\rangle) = y] = 2^{-m}$ always, which completes the proof. □

4.2 The Statistical Setting

The statistical setting is more nuanced than the perfect setting, but is still governed by a great deal of structure. In this section we will completely state the relationships among the five statistical varieties of functions that we have introduced thus far. After doing so, we will introduce one more class of functions—*almost perfect resilient functions*—that is strictly stronger than each of the five other types.

Let us first establish some notational conventions: we will write $\mathcal{WSERF}$ to denote the set of triples (f, k, ε) where f is a weakly static statistical (k, ε)-ERF, $\mathcal{SAERF}$ to denote the analogous set for strongly adaptive statistical ERF's, and so on. Using this notation, we may write, for instance, $\mathcal{SSERF} \subseteq \mathcal{WSERF}$ to mean that for all settings of parameters (n, m, k, ε), any function f that is a strongly static ERF with those parameters is also a weakly static ERF with the same parameters.

4.2.1 Inclusions

We begin by noting the obvious inclusions: $\mathcal{SSERF} \subseteq \mathcal{WSERF}$, $\mathcal{SAERF} \subseteq \mathcal{WAERF}$, $\mathcal{SAERF} \subseteq \mathcal{SSERF}$, and $\mathcal{WAERF} \subseteq \mathcal{WSERF}$. Where does $\mathcal{RF}$ fit in? It is clear that if setting the bits of the input doesn't help the adversary, then merely accessing them surely won't, so we know that

$\mathcal{RF} \subseteq \mathcal{WSERF}$. However, the reduction is less obvious if we give an adversary *adaptive* read-only access to the input of an RF. In fact, the inclusion still holds, and our alternate characterization makes the reason for this clear.

Lemma 4.2. *Let $f\colon \{0,1\}^n \to \{0,1\}^m$ be a statistical (k,ε)-RF. Then f is a weakly adaptive statistical (k,ε)-ERF. That is, $\mathcal{RF} \subseteq \mathcal{WAERF}$.*

Proof. By the alternate definition of weakly adaptive ERF's (Proposition 3.11), we need to show that, for every branching function $\varphi\colon \{0,1\}^{n-k} \to \left\{{n \atop n-k}\right\}$, f satisfies

$$\mathop{\mathrm{E}}_{a \leftarrow \langle U_{n-k}\rangle}[\Delta(f(\langle \varphi(a)^{a,n}\rangle), \langle U_m\rangle)] \leq \varepsilon$$

But since f is a statistical (k,ε)-RF, $\Delta(f(\langle \varphi(a)^{a,n}\rangle), \langle U_m\rangle) \leq \varepsilon$ always, which implies the result. □

Lemma 4.2 holds simply because RF's extract from *all* (n,k) oblivious bit-fixing sources, and even though weakly adaptive ERF's are a bit trickier than static ones, they still are essentially just average-case extractors of sorts for (n,k) oblivious bit-fixing sources.

The inclusions we've discussed are summarized in Figure 4.1. In the figure, existence of a path from class $\mathcal{A}$ to class $\mathcal{B}$ implies that $\mathcal{A} \subseteq \mathcal{B}$.

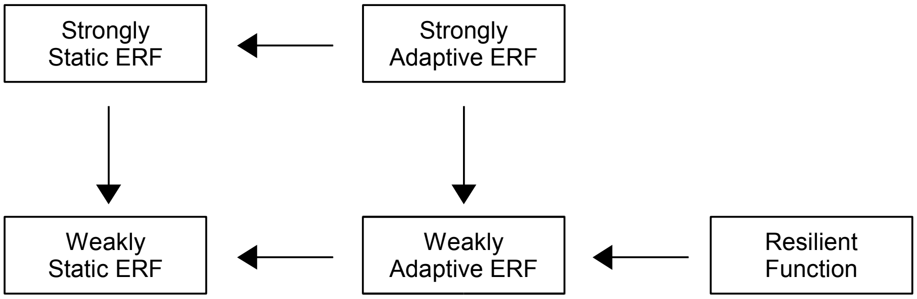

FIGURE 4.1: Relationships among statistical RF's and ERF's.

Thus far, we have merely proven four trivial inclusions as well as a fifth inclusion that was similarly simple. Is this really the whole story? In other words, do any of the inclusions not yet considered (besides the ones implied by Figure 4.1) hold? It turns out that none do, as we show in the next section by proving that it is impossible to add any arrow with a non-zero rightward or upward component to Figure 4.1.

4.2.2 Separations and a Preliminary Classification

Our goal in this section will be to prove the necessary separations to show that Figure 4.1 is "complete" in the sense that $\mathcal{A} \subseteq \mathcal{B}$ if *and only if* there is a path in the diagram from $\mathcal{A}$ to $\mathcal{B}$.

All of the constructions that we exhibit for our separations are polynomial-time computable, ruling out the possibility that the separations do not hold when we restrict our classes to contain only polynomial-time computable functions.

Three Separations

We begin by showing that $\mathcal{SSERF} \not\subseteq \mathcal{WAERF}$ with a simple construction: our function will return the exclusive-or of approximately half of its input bits, but it will use a signal bit at the end of its input to decide exactly *which* half of the bits it will use. This offers no help to the strongly static adversary even if that adversary is given access to all but two of the input bits, but it is enough to give a weakly adaptive adversary considerable advantage even for k as large as $(n-1)/2$.

Lemma 4.3. *For every odd $n \geq 3$, there exists a function $f\colon \{0,1\}^n \to \{0,1\}$ that is polynomial-time computable and is a strongly static statistical $(2, 1/4)$-ERF but not a weakly adaptive statistical (k, ε)-ERF for any $k \leq (n-1)/2$ and $\varepsilon < 1/2$.*

Proof. For odd n, define $f\colon \{0,1\}^n \to \{0,1\}$ as follows: Let $S = [\frac{n-1}{2}]$, and let $T = \{\frac{n+1}{2}, \ldots, n-1\}$ (each of these sets is of size $\frac{n-1}{2}$). On an input $w \in \{0,1\}^n$, check if $w_n = 0$. If so, return the exclusive-or of the bits of $[w]_S$. Otherwise return the exclusive-or of the bits of $[w]_T$.

Let $k \leq \frac{n-1}{2}$. The function f is not a weakly adaptive statistical (k, ε)-ERF for any $\varepsilon < 1/2$, because given access to a randomly chosen input $r \leftarrow \langle U_n \rangle$, an adaptive adversary can request $[r]_n$, use that knowledge to request either $[r]_S$ or $[r]_T$, and then compute f with complete accuracy, giving it an advantage of $1/2$ in distinguishing $f(r)$ from a uniformly random bit.

We now prove, on the other hand, that f is a strongly static statistical $(2, \varepsilon)$-ERF for any $\varepsilon \geq 1/4$. For $\beta \in \{0,1\}$, let L_β be the set that the (strong) adversary chooses to expose if it receives β as the challenge; that is, β is the one-bit-long string for which the adversary is trying to distinguish whether it came from $f(\langle U_n \rangle)$ or $\langle U_m \rangle$. The alternate characterization of strongly static statistical ERF's (Proposition 3.12) gives that we need f to satisfy the following equation. (We've used linearity of expectation to switch the expectation and the sum.)

$$\frac{1}{2} \sum_{\beta \in \{0,1\}} \left[\mathop{\mathrm{E}}_{a \leftarrow \{0,1\}^{n-2}} \left| \Pr[f(\langle L_\beta^{a,n} \rangle) = \beta] - \frac{1}{2} \right| \right] \leq \varepsilon \tag{4.1}$$

But this is easy to bound: $\left| \Pr[f(\langle L_\beta^{a,n} \rangle) = \beta] - \frac{1}{2} \right|$ is equal to 0 unless either $S \subset L_\beta$ or $T \subset L_\beta$, so we may assume without loss of generality that $S \subset L_\beta$. This leaves $\frac{n-1}{2} - 1$ more bits for us to put in L_β. Including in L_β any bits from T without including *all* of T will produce a distribution $f(\langle L_\beta^{a,n} \rangle)$ that is identical to the one produced when L_β includes no bits from T. This is because the output of f, if it uses any bits in T, will take the exclusive-or of *all* the bits in T. However, there is no way to include all of the bits of T in L_β since the adversary can only choose $\frac{n-1}{2} - 1$ more bits, so we may assume without loss of generality that $L_\beta = [n] - \{n-1, n-2\}$.

In this case, for all values of a whose last bit is 0, we have that $f(\langle L_\beta^{a,n} \rangle)$ is a degenerate distribution on either $\{0\}$ or $\{1\}$; either way, $\left|\Pr[f(\langle L_\beta^{a,n} \rangle) = \beta] - \frac{1}{2}\right| = \frac{1}{2}$. For the values of a whose last bit is 1, $f(\langle L_\beta^{a,n} \rangle)$ is a uniformly random bit, and so $\left|\Pr[f(\langle L_\beta^{a,n} \rangle) = \beta] - \frac{1}{2}\right| = 0$. Thus, the expectation over a of $\left|\Pr[f(\langle L_\beta^{a,n} \rangle) = \beta] - \frac{1}{2}\right|$ is 1/4, implying that Equation 4.1 holds for any $\varepsilon \geq 1/4$, as claimed. □

This result is useful for our project, but we need more. The following lemma will establish that $\mathcal{RF} \not\subseteq \mathcal{SSERF}$. The function we build here essentially uses the second part of its input to choose a subset S of the bits from the first part of the input and computes the exclusive-or of the bits in S. It then outputs a description of S along with the result of the exclusive-or computation. Any strong adversary will know from the output which subset of the input bits it should request in order to calculate f, and so f will fail to be a strongly static ERF. But why is the resultant function an RF? The reason is that we don't take the second part of the input as it is in order to decide which bits from the first part to include in the set S; instead, we first apply a resilient function to the second part, thus obfuscating the information required to bias the output of the function. This technique gives us an exponential separation in the error parameter ε.

Lemma 4.4. *For every constant $0 < \gamma < 1/2$ and for infinitely many $n \in \mathbb{N}$, there exists a polynomial-time computable function $f\colon \{0,1\}^n \to \{0,1\}^m$ and a function $0 < k \leq n$ such that f is a statistical $\left(k, 2^{-\Omega(n^\gamma)}\right)$-RF but, for any $\varepsilon < 1/4 - 2^{-\Omega(n^\gamma)}$, f fails to be a strongly static statistical (k,ε)-ERF.*

Proof. This takes some work to show, so we will explicitly split our proof into three parts. We will first state our definition of f (Part 1), then prove that f fails to be a strongly static statistical ERF (Part 2), then prove that it is a statistical RF (Part 3),

Part 1. Definition of f.
Using Theorem 2.15 (construction of a deterministic extractor for oblivious bit-fixing sources) and Proposition 3.8 (alternate characterization of statistical RF's), we have for all $n' \in \mathbb{N}$ a polynomial-time computable statistical (k', ε')-RF f' with the following parameters:

1. Input length: n'
2. Output length: $m' = n'/2 - {n'}^{1/2+\gamma}$
3. Protected bits: $k' = n'/2$
4. Error: $\varepsilon' \in 2^{-\Omega({n'}^\gamma)}$

We now define f as follows: Given an input $w \in \{0,1\}^{m'+n'}$, split w into its first m' bits (call these w_1) and its remaining n' bits (call these w_2). Apply f' to w_2 to obtain a string $s \in \{0,1\}^{m'}$.

Interpret s as a subset $S \subseteq [m']$ by including the i-th element of $[m']$ in S if and only if the i-th bit of s is 1. Set the output of f to be $(s, \bigoplus S) \in \{0,1\}^{m'+1}$ where $\bigoplus S$ denotes the exclusive-or of the bits of $[w_1]_S$.

We have defined f only for input lengths of the form $m' + n' = 3n'/2 - {n'}^{1/2+\gamma}$. On inputs of this length, which we write below as functions of n', f has the following parameters:

1. Input length: $n(n') = 3n'/2 - {n'}^{1/2+\gamma}$
2. Output length: $m(n') = m'(n') + 1 = n'/2 - {n'}^{1/2+\gamma} + 1$
3. Protected bits: we will use $k(n') = n - (m(n') - 1)/2$

For the remainder of the proof, we suppress the parametrization by n'.

Part 2. f fails to be a strongly static statistical (k, ε)-ERF with $\varepsilon < 1/4 - 2^{-\Omega(n^\gamma)}$ for infinitely many n.
Let $n = m' + n'$ be the length of an input. In this case, an adversary with access to $(s, \bigoplus S)$ can use s to request the set S of the input bits and calculate $\bigoplus S$, thus giving it advantage of $1/2$ in distinguishing between $f(\langle U_n \rangle)$ and $\langle U_{m'+1} \rangle$. However, S is a subset of $[m'] = [m-1]$, and the adversary can only do this if $|S| \leq (m-1)/2$ since otherwise it would have to request too many bits. Fortunately, the probability of a randomly chosen subset of $[m-1]$ having at most $(m-1)/2$ elements is at least $1/2$ since the number of subsets of size i equals the number of subsets of size $m-1-i$. Furthermore, f', which calculates S, carries the uniform distribution to a distribution that is ε'-close to uniform. Therefore, the adversary will be able to request all the bits of S at least a $1/2 - \varepsilon'$ fraction of the time. Noting that $n, m \in O(n')$, and that therefore $\varepsilon' \in 2^{-\Omega(n^\gamma)}$, we see that an adversary can always gain an advantage of at least $(1/2 - \varepsilon')/2 \geq 1/4 - 2^{-\Omega(n^\gamma)}$.

Part 3. f is a statistical $\left(k, 2^{-\Omega(n^\gamma)}\right)$-RF for sufficiently large n
The intuition here is that since no adversary can set enough bits to bias f', no adversary can get much information about S and therefore the output of f will appear random. Let us write the argument rigorously.

Let $n = m' + n'$ be the length of an input. Now fix any $L \in \left\{{n \atop n-k}\right\}$ and $a \in \{0,1\}^{n-k}$ and let $L_1, L_2 \subset [n]$ and $a_1, a_2 \in \{0,1\}^*$ be the unique pair of sets and pair of strings such that $[L^{a,n}]_{[m']} = L_1^{a_1,m'}$ and $[L^{a,n}]_{[n']+m'} = L_2^{a_2,n'}$. We will exploit the fact that our statistical distance measure Δ obeys the triangle inequality (see Lemma 2.6 from Chapter 1) to bound the distance between $f(\langle L^{a,n} \rangle)$ and $\langle U_{m'+1} \rangle$. We do this by bounding

$$\rho = \Delta\left(f(\langle L^{a,n} \rangle), \left(f'(\langle L_2^{a_2,n'} \rangle), \langle U_1 \rangle\right)\right)$$

and

$$\delta = \Delta\left(\left(f'(\langle L_2^{a_2,n'} \rangle), \langle U_1 \rangle\right), \langle U_{m'+1} \rangle\right) = \Delta\left(f'(\langle L_2^{a_2,n'} \rangle), \langle U_{m'} \rangle\right)$$

Bounding δ is easy: since $n-k \leq m'/2$, we can assume $|L| \leq m'/2 < n'/2$, implying that $|L_2| < n'/2$. Since f' is a statistical $(n'/2, \varepsilon')$-RF, this gives that $f'(\langle L_2^{a_2,n'}\rangle) \cong_{\varepsilon'} \langle U_{m'}\rangle$, meaning that $\delta \leq \varepsilon'$.

We now bound ρ: For $w \in \{0,1\}^{m'}$, let $P_w = \Pr[f'(\langle L_2^{a_2,n'}\rangle) = w]$ and let W denote the subset of $[m']$ that corresponds to w. Lemma 2.6 from Chapter 1, together with the fact that $f'(\langle L_2^{a_2,n'}\rangle)$ is the first m' bits of $f(\langle L^{a,n}\rangle)$, allows us to write:

$$\begin{aligned}\rho &= \mathop{\mathrm{E}}_{w \leftarrow f'(\langle L_2^{a_2,n'}\rangle)}\left[\Delta\left(\bigoplus_{i\in W}[\langle L_1^{a_1,m'}\rangle]_i, \langle U_1\rangle\right)\right] \\ &= \sum_{w\in\{0,1\}^{m'}} P_w \cdot \left|\Pr\left[\bigoplus_{i\in W}[\langle L_1^{a_1,m'}\rangle]_i = 0\right] - \frac{1}{2}\right| \qquad (4.2)\end{aligned}$$

Note that in the second line we compute Δ without summing over $\beta \in \{0,1\}$ and multiplying by $1/2$. This is because the summand would be the same for $\beta = 0$ and $\beta = 1$. Note further that if W is contained in L_1 then $\bigoplus_{i\in W}[\langle L_1^{a_1,m'}\rangle]_i$ is the exclusive-or of bits that are all fixed, so it is a degenerate distribution. On the other hand, if W is *not* contained in L_1, some of the bits in $[\langle L_1^{a_1,m'}\rangle]_W$ will be uniformly random and, consequently, $\bigoplus_{i\in W}[\langle L_1^{a_1,m'}\rangle]_i$ will be uniformly random as well. Thus, the summand in Equation 4.2 is equal to $1/2$ for $W \subset L_1$ and 0 otherwise. How many strings w give $W \subset L_1$? One for each subset of L_1; that is, $2^{|L_1|} \leq 2^{n-k} < 2^{m'/2}$ strings. Letting B denote this set of "bad" strings w that give $W \subset L_1$, we can re-write Equation 4.2 as follows.

$$\begin{aligned}\rho &= \sum_{w\in B} P_w \cdot \left|\Pr\left[\oplus_{i\in W}[\langle L_1^{a_1,m'}\rangle]_i = 0\right] - \frac{1}{2}\right| \\ &= \sum_{w\in B} P_w \cdot \frac{1}{2} \\ &< \frac{1}{2}\Pr\left[f'(\langle L_2^{a_2,n'}\rangle) \in B\right] \\ &< \frac{1}{2}\cdot\left(\frac{2^{m'/2}}{2^{m'}} + \varepsilon'\right) \qquad (4.3) \\ &= 2^{-m'/2} + \frac{\varepsilon'}{2}\end{aligned}$$

Where line 4.3 follows from the fact that $f'(\langle L_2^{a_2,n'}\rangle) \cong_{\varepsilon'} \langle U_{m'}\rangle$.

Having bounded ρ, we can now finally bound $\Delta(f(\langle L^{a,n}\rangle), \langle U_{m'+1}\rangle)$, the distance of interest:

$$\begin{aligned}\Delta(f(\langle L^{a,n}\rangle), \langle U_{m'+1}\rangle) &\leq \delta + \rho \\ &< \varepsilon' + \left(2^{-m'/2} + \frac{\varepsilon'}{2}\right) \\ &= \frac{3}{2}\cdot\varepsilon' + 2^{-m'/2}\end{aligned}$$

Recalling that $\varepsilon' \in 2^{-\Omega(n^\gamma)}$ and that $m'(n') \in \Omega(n(n')) \subset \Omega(n(n')^\gamma)$, we see that we have indeed shown that f is a statistical (k, ε)-RF with $\varepsilon \in 2^{-\Omega(n^\gamma)}$. □

The last lemma we will need is the separation $\mathcal{SAERF} \not\subseteq \mathcal{RF}$, which is a strengthening of the converse of Lemma 4.4. In this construction, we take a perfectly good SAERF (which may very well also be an RF) and "sabotage" its worst-case behavior without hurting its average-case behavior too much. Recall that we showed in Section 3.2 that RF's require good wost-case behavior while ERF's require only good average-case behavior. By this intuition our modified function will no longer be resilient; on the other hand, since we will not have affected the original function's average-case behavior too much, it will remain a strongly adaptive statistical ERF even after our modification. As in the previous lemma, this achieves a separation that is exponential in terms of the error parameter ε.

Lemma 4.5. *Every strongly adaptive statistical (k, ε')-ERF $f' \colon \{0,1\}^n \to \{0,1\}^m$ can be modified in polynomial time to give a function $f \colon \{0,1\}^n \to \{0,1\}^m$ that is a strongly adaptive statistical (k, ε)-ERF for $\varepsilon = \varepsilon' + 2^{-k}$ but not a statistical (k, ε)-RF for any $\varepsilon < 1 - 2^{-m}$.*

Proof. Let f' be any strongly adaptive statistical (k, ε')-ERF. We build f as follows: let $S = [n-k]$; now, on an input $w \in \{0,1\}^n$, we set $f(w) = f'(w)$ unless $[w]_S = 0^{n-k}$ in which case we set $f(w) = 0^m$.

f is not a statistical (k, ε)-RF for any $\varepsilon < 1 - 2^{-m}$ because $f(\langle S^{0^{n-k},n} \rangle)$ is the degenerate distribution on $\{0^m\}$ and so is at a distance of $1 - 2^{-m}$ from $\langle U_m \rangle$.

On the other hand, f is a strongly adaptive statistical (k, ε)-ERF for $\varepsilon = \varepsilon' + 2^{-k}$ for the following reason: fix any function $d \colon \{0,1\}^n \times \{0,1\}^m \to \left\{ {n \atop n-k} \right\}$ that reads at most $n-k$ bits of its first input, and define the following two distributions:

$$\begin{aligned} A(\langle U_n \rangle) \quad &:= \big([\langle U_n \rangle]_{d(\langle U_{n,} \rangle f(\langle U_n \rangle)))}, f(\langle U_n \rangle) \big) \\ B(\langle U_n \rangle) \quad &:= \big([\langle U_n \rangle]_{d(\langle U_{n,} \rangle f'(\langle U_n \rangle)))}, f'(\langle U_n \rangle) \big) \end{aligned}$$

As our notation suggests, $A(\langle U_n \rangle)$ and $B(\langle U_n \rangle)$ are both ultimately just deterministic functions of $\langle U_n \rangle$, and they agree whenever f and f' agree. In particular, A and B are exactly identical on $\{0,1\}^n - S^{0^{n-k},n}$. We can conclude from this that $A(\langle U_n \rangle)$ and $B(\langle U_n \rangle)$ will be as different as possible if $A(\langle S^{0^{n-k},n} \rangle)$ and $B(\langle S^{0^{n-k},n} \rangle)$ are each equal to a different one-element set. But even if this is true, the distance between $A(\langle U_n \rangle)$ and $B(\langle U_n \rangle)$ is at most 2^{-k}. Now since f' is a strongly adaptive statistical (k, ε')-ERF, $B(\langle U_n \rangle) \cong_{\varepsilon'} \langle U_m \rangle$ and so by the triangle inequality for Δ we have that $\Delta(A(\langle U_n \rangle), \langle U_m \rangle) \leq \varepsilon' + 2^{-k}$, completing the proof. □

Remark 4.6. This establishes the separation that we desire *only* provided that we prove that there exists a strongly adaptive statistical ERF f' (for infinitely many n of course). Further, the separation is for polynomial-time computable functions and is exponential, as claimed, provided that f' is polynomial-time computable and has an exponentially small error parameter ε'. We will construct such a function in Section 4.2.3 after we are finished with our classification.

Beginning the Classification

It is straightforward to verify that the three separations above are all we need to prove that Figure 4.1 is complete and that indeed no other arrows can be drawn on it.

Theorem 4.7. *Figure 4.1 is complete in the sense that a path from $\mathcal{A}$ to $\mathcal{B}$ exists if and only if $\mathcal{A} \subseteq \mathcal{B}$.*

Proof. The existence of any arrow beyond the ones on the diagram contradicts at least one of Lemmas 4.3, 4.4, and 4.5. □

Theorem 4.7 finally gives us the reason that we could not characterize the strong ERF's in terms of average-case randomness extraction from (n, k) oblivious bit-fixing sources in Section 3.2: if we managed to define the strong ERF's in terms of their ability to extract from some family of (n, k) oblivious bit-fixing sources, then every RF would, by virtue of its ability to extract from *all* (n, k) OBFS's, become a strong ERF as well, but we now know that this would be a contradiction!

The classification at which we have arrived is quite pleasing. However, there is something unsettling about it: it has left us with no "strongest" object, for RF's are not stronger than SAERF's and the latter are not stronger than the former either. Is there a class of functions that we can put at the top of our hierarchy?

4.2.3 Almost-Perfect RF's and Classifying the Entire Zoo

As the title of this section may have betrayed, the answer to the question we have posed is "yes". But how can we gain intuition about how to find the family of functions that we seek? As we noted in Remark 3.13 following our alternate characterization of RF's and ERF's, the resilient functions are a sort of worst-case analogue of the weak ERF's, so perhaps we can come up with a corresponding worst-case analogue for the *strong* ERF's. This would need to be a sub-class of the RF's (since strong ERF's are a subclass of weak ERF's) that is a sub-class of the strongly adaptive ERF's in the same way that the RF's are a sub-class of the weakly adaptive ERF's.

In the world of the weak ERF's, the expression that we average in the alternate definitions of Section 3.2 is $\Delta(f(\langle L^{a,n}\rangle), \langle U_m\rangle)$, and we average over different types of sets L and strings a depending on whether f is an adaptive or static weak ERF. To turn such a function into an RF we simply stipulate that it satisfy this condition for *every* set L and string a. Let us apply this reasoning to our average-case definition of the strong ERF's.

Our alternate characterizations of static and adaptive strong ERF's both involve averaging expressions of the form

$$\sum_{z \in \{0,1\}^m} \left| \Pr[f(\langle {L_{z,a}}^{a,n} \rangle) = z] - 2^{-m} \right| \tag{4.4}$$

where we average over $a \in \{0,1\}^{n-k}$, and various families of sets depending on z and possibly a take the place of $L_{z,a}$. When attempting to turn this into a worst-case condition, we could consider simply requiring that Expression (4.4) be at most ε, but then it is not obvious that such a function would be a statistical RF, which is what we seek from the definition we are trying to formulate. However, if we strengthen our demands by requiring that the summand in Expression (4.4) be at most $\varepsilon/2^m$ for all L, z, and a, we ensure that the resultant function will be an RF because substituting any fixed L for $L_{z,a}$ in Expression (4.4) just turns the expression into $\Delta(f(L^{a,n}), \langle U_m \rangle)$, and if its summand is at most $\varepsilon/2^m$ then the expression itself (which sums over 2^m elements) is at most ε.

The object we have arrived at is called an *almost-perfect resilient function* (APRF) because beyond asking that $f(\langle L^{a,n} \rangle)$ be ε-close to uniform for all L and a, we've essentially asked that the deviation from uniform be very evenly distributed (and therefore very small) across all possible $z \in \{0,1\}^m$. Let us formalize our above line of reasoning by giving a definition of APRF's and proving that $\mathcal{APRF} \subseteq \mathcal{RF}$ and $\mathcal{APRF} \subseteq \mathcal{SAERF}$.

Definition 4.8 (Almost-Perfect Resilient Function)**.** A function $f\colon \{0,1\}^n \to \{0,1\}^m$ is called a (k,ε)*-APRF* if and only if for every $L \in \left\{ {n \atop n-k} \right\}$, $a \in \{0,1\}^{n-k}$, and $z \in \{0,1\}^m$, we have:

$$\left| \Pr[f(\langle L^{a,n} \rangle) = z] - 2^{-m} \right| \leq \frac{\varepsilon}{2^m}$$

Lemma 4.9. *Let $f\colon \{0,1\}^n \to \{0,1\}^m$ be a (k,ε)-APRF. Then f is both a statistical $(k,\varepsilon/2)$-RF and a strongly adaptive statistical $(k,\varepsilon/2)$-ERF.*

Proof. f is a statistical $(k,\varepsilon/2)$-RF because

$$\begin{aligned} \Delta\left(f(\langle L^{a,n} \rangle), \langle U_m \rangle\right) &= \frac{1}{2} \sum_{z \in \{0,1\}^m} \left| \Pr\left[f(\langle L^{a,n} \rangle) = z\right] - \frac{1}{2^m} \right| \\ &\leq \frac{1}{2} \sum_{z \in \{0,1\}^m} \frac{\varepsilon}{2^m} \\ &= \frac{\varepsilon}{2} \end{aligned}$$

To show that f is a strongly adaptive statistical $(k,\varepsilon/2)$-ERF, we proceed in the same way that we showed that every RF is a weakly adaptive ERF (Lemma 4.2). We know by the adaptive variant of the definition of strongly adaptive ERF's (Proposition 3.12) that it suffices to show that for every function $d\colon \{0,1\}^{n-k} \times \{0,1\}^m \to \left\{ {n \atop n-k} \right\}$ that is a branching function in its first argument, f satisfies:

$$\frac{1}{2} \operatorname*{E}_{a \leftarrow \langle U_{n-k} \rangle} \left[\sum_{z \in \{0,1\}^m} \left| \Pr[f(\langle d(a,z)^{a,n} \rangle) = z] - 2^{-m} \right| \right] \leq \varepsilon$$

From here, simply plugging in $|\Pr[f(\langle d(a,z)^{a,n} \rangle) = z] - 2^{-m}| \leq \varepsilon/2^m$ gives that the inequality holds as desired. □

In this lemma, as with Lemma 4.2, our alternate characterization makes plain the reason that the reductions from resilience to exposure-resilience are true: both cases are simply a matter of worst-case performance implying average-case performance.

The Complete Map of the RF and ERF Zoo

We are now in a position to state the *whole* story: an extension of Theorem 4.7 that includes the "strongest object" in our zoo.

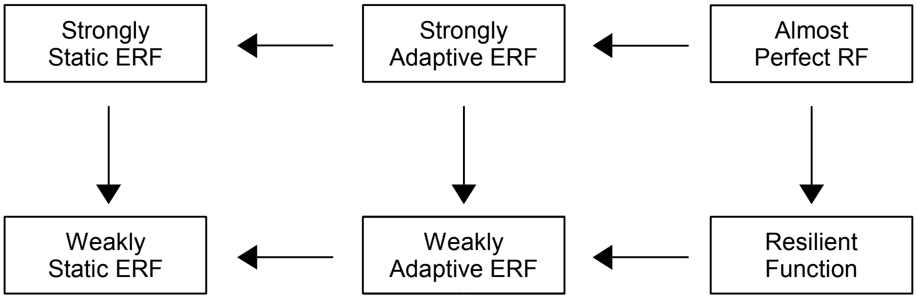

Figure 4.2: Relationships among statistical RF's and ERF's, including APRF's.

Theorem 4.10. *Figure 4.2 is a complete classification of the (statistical) RF's and ERF's in the sense that a path from $\mathcal{A}$ to $\mathcal{B}$ exists if and only if $\mathcal{A} \subseteq \mathcal{B}$.*

Proof. The separation between $\mathcal{RF}$ and $\mathcal{SAERF}$, together with Lemma 4.9, suffices to extend Theorem 4.7 to include Figure 4.2 □

We have successfully organized our zoo into something quite neat: all of the "perfect" animals turned out to be perfectly identical, and the statistical ones behave according to a very nice hierarchy. We should be careful to note, however, that our separations only hold if we force k and ε to remain fixed for any potential reduction, and they do not necessarily hold when we remove this condition: for example, every (k, ε)-RF is trivially a $(k, 2^m\varepsilon)$-APRF. This may seem unimportant because of the exponential factor in the error term, but in fact it is the key to constructing a usable APRF, as we see in the next section.

Real, Live APRF's

We have found in APRF's a good candidate for a "strongest" class of functions to include in our classification, but we have not yet shown that any APRF's even exist!

The construction of an APRF offers us a few rewards at this point. First of all, our classification shows that constructing an APRF would give us an SAERF, which would make concrete the strong separation of Lemma 4.5 (see Remark 4.6). But in addition to giving us a separation result, constructing APRF's is actually the best currently known way of constructing SAERF's, and therefore adaptive all-or-nothing transforms (see Section 3.3.4). For this second reason, APRF's are of extreme practical importance as well as theoretical interest.

Dodis, Sahai, and Smith [21] present a probabilistic construction of (k, ε)-APRF's with very good parameters ($m = k - 2\log(1/\varepsilon) - O(\log n)$) by choosing a random function out of a t-wise independent hash family (such as, for example, the set of polynomials of degree $t-1$ over $\mathbb{F}_{2^n}$).[1] This construction is efficiently computable and almost always yields a good APRF, but it is not uniform: a different function from a different family needs to be chosen and stored for every input-length n. Thus, we cannot use it to construct a polynomial-time computable APRF for infinitely many n.

However, as we hinted in the previous section, it actually turns out that there are deterministic extractors for oblivious bit-fixing sources that are good enough to be used directly as APRF's. One of these is the extractor of Kamp and Zuckerman [12] cited in Theorem 2.17 of the preliminaries. This elegant construction uses special graphs called expander graphs that are well-connected despite having relatively few edges. Because of this connectivity, a random walk on an expander graph with 2^m vertices is likely to yield a nearly random endpoint (that can be interpreted as an element of $\{0,1\}^m$), just as a random walk on the complete graph on 2^m vertices would give a completely random endpoint. For this reason, randomness extractors are strongly related to expanders and there is a large body of literature that states and exploits connections between these objects.

Kamp and Zuckerman [12] use a connection similar (but not identical) to this one to construct a deterministic extractor by showing a way to convert an oblivious bit-fixing source into a source that is "good enough" to serve as the source of randomness for a random walk on an expander graph.[2] This yields, as stated in Theorem 2.17, a deterministic extractor for (n, k) oblivious bit-fixing sources from n bits to $\Omega(n^{2\gamma})$ bits with $k = n^{\gamma+1/2}$ and $\varepsilon < 2^{-cm}$ (for any $c > 1$ and $\gamma \leq 1/2$), which as we mentioned above is trivially an APRF with $\varepsilon = 2^{-(c-1)m}$. The result is the following theorem, which asserts the existence of APRF's for $k \gg \sqrt{n}$.

Theorem 4.11 (Kamp and Zuckerman [12]). *For every constant $0 < \gamma \leq 1/2$ and every constant $c > 0$, and for every n and k satisfying $n \geq k \geq n^{1/2+\gamma}$, there exists a polynomial-time computable (k, ε)-APRF $f\colon \{0,1\}^n \to \{0,1\}^m$ with $m \in \Omega(n^{2\gamma})$ and $\varepsilon \leq 2^{-cm}$.*

[1] A t-wise independent hash family is a collection of functions such that, for every t distinct inputs $\{x_1, \ldots, x_t\}$, the t random variables produced by choosing a random function from the family and applying it to the x_i behave like uniformly distributed, independent random variables.

[2] The connection they exploit is a new one because their walk is not a random walk but rather a walk with some random steps and some steps that are biased ahead of time. (The latter steps arise from the fixed bits of the source.)

4.3 References

It is doubtful that the conclusions of Theorem 4.1 escaped the notice of other authors (and in particular Dodis, Sahai, and Smith mention a few of them in [21]). However, its explicit statement is original, as is the given proof which uses the original alternate characterizations of Section 3.2. Dodis, Sahai, and Smith (also in [21]) state the complete classification of ERF's, RF's, and APRF's and so none of the inclusions and separations proven in the statistical setting are original results. However, the proofs of all the inclusions and separations except for that of Lemma 4.4 are original either in that the lemmas' statements elsewhere are given less explicitly and without proof, or that the proofs use the alternate characterizations from Section 3.2, or both. In the case of Lemma 4.4, we have filled in the gaps of a proof sketch and made the statement of the lemma more precise in terms of the parameters given. APRF's were introduced by Kurosawa, Johansson, and Stinson [10] prior to their use in [21].

Chapter 5

Resilience Amplification

In this chapter we consider a function $f\colon \{0,1\}^n \to \{0,1\}^m$ that is a (k, ε)-RF or ERF of some sort and ask whether its resilience can somehow be "amplified". We first define what we might mean by this, then focus on one specific interpretation (increasing f's input length without changing k) in the case where f is a weakly static ERF. We define a natural class of candidate functions—lossless condensers for WSERF's—for achieving this form of amplification and then rule out their existence in both the perfect and statistical settings.

In our analysis, we will make heavy use of the convention that if $\langle X\rangle$ is a flat distribution then X is implicitly defined to be its support. Similarly, if X is a set then $\langle X\rangle$ is implicitly defined to be the uniform distribution on X. (See Section 2.1.2 from the preliminaries.) Our results also use some basic facts about oblivious bit-fixing sources that are discussed in Appendix B.

5.1 Background and Motivation

Before investigating resilience amplification in earnest, we will first make precise our notion of what it means in the context of RF's and ERF's. After doing so, we will discuss the utility of amplification techniques as objects of study, mention previous uses of similar concepts, and investigate how we might apply resilience amplification to weakly static ERF's.

5.1.1 Resilience Amplification and Its Benefits

Consider a (k, ε)-RF or ERF $f \colon \{0,1\}^n \to \{0,1\}^m$. There are two natural ways in which we might go about improving f that correspond to our two goals in building it:

- We can try to decrease the ratio k/n (the fraction of the input bits that requires protection in order for f to work).
- We can try to increase m, the number of pseudorandom bits that f outputs.

The existence of a procedure that would achieve either of these two goals would be useful for two reasons. First, such a procedure might allow us to improve the parameters of existing constructions. Second, since few bounds on the achievable parameters of RF's and ERF's are known (see the discussion in Section 5.1.3), such a procedure could offer insight into the kinds of bounds that we can expect to prove about these functions.

5.1.2 Previous Uses of Amplification Procedures

Both of the aforementioned amplification procedures have proven fruitful when applied to other pseudorandom objects. In particular, they have significantly improved constructions of both pseudorandom generators and seeded randomness extractors.

Lengthening Output: PRG's

The discovery of a general procedure that takes a PRG from n bits to $n + 1$ bits and extends its output length to any polynomial in n (see Goldreich [19]) showed that stretching randomness by just one bit was as hard as stretching it into polynomially many bits. This amplification method has resulted in polynomial-time computable constructions of many types of PRG's (including the PRG construction of Hastad et al. [18] cited in Theorem 2.25) that achieve polynomial stretch.

Handling Lower Quality Randomness: Seeded Extractors

As is the case with RF's and ERF's, one of the parameters of interest for a seeded randomness extractor for k-sources $E \colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ is k/n, the so-called *min-entropy rate* that E can handle. This parameter is important because the higher the min-entropy rate of a source, the easier it is to extract randomness from that source. Thus, the construction of good extractors was made much easier following the introduction and construction of functions that convert a source of low min-entropy rate into one of high min-entropy rate. Such functions are called *condensers* and are formally defined below.

Definition 5.1 (Condenser for Seeded Extractors[1]). A function $C\colon \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m$ is a $k \to_\varepsilon k'$ *condenser for seeded extractors* if and only if, for every k-source $\langle X \rangle$ on $\{0,1\}^n$, $C(\langle U_s \rangle, \langle X \rangle) \approx_\varepsilon \langle Y \rangle$ for some k'-source $\langle Y \rangle$.

As we have mentioned before, a result of Radhakrishnan and Ta-Shma [24] shows that condensers for seeded extractors with $m = k'$ (which are just seeded extractors) must suffer a minimum entropy loss expressed by $k + s - m \geq 2\log(1/\varepsilon) + O(1)$. However, condensers for seeded extractors with general parameters (i.e. potentially having $m > k'$) need not have this entropy loss. Condensers for seeded extractors that have $k' = k + s$ and so produce no entropy loss *at all* are called *lossless condensers for seeded extractors*. The construction of lossless condensers for seeded extractors with good parameters has been crucial to the theory of randomness extractors, yielding the best known constructions of extractors for sources with very low min-entropy rate (including the construction of Guruswami, Umans, and Vadhan [17] cited in Theorem 2.21, which is the current state of the art).

5.1.3 Application to Weakly Static ERF's

Can we use amplification methods to learn about RF's and ERF's? To make this question more manageable, let us narrow our focus and suppose for the rest of this chapter that the function f whose properties we are attempting to amplify is a weakly static (perfect, statistical, or computational) ERF.

An Open Question

Recall that in Section 3.4.2 we presented a weakly static statistical ERF that had essentially optimal output length ($m = \gamma k$ for any constant $0 < \gamma < 1$) but required the number of protected bits to be $k \in \omega(\log n)$; we also presented a weakly static statistical ERF that had *no* restrictions on k but only achieved output length $m = (\log k)/4$. The fact that no known construction achieves both of these goals simultaneously suggests that there may be a fundamental trade-off between k/n (the fraction of bits that require protection) and the output length m. In the perfect setting, we have a result that indeed describes such a trade-off: Chor et al. [23] show that $m \geq 2$ necessitates $k \geq n/3$. However, it is not known whether this sort of trade-off is necessary in the statistical setting.

Currently, the largest value of m for which we have weakly static statistical (k,ε)-ERF's with constant k and $\varepsilon \leq 1/2$ is $O(\log k)$, achieved by the aforementioned construction of Theorem 3.20. Can we achieve $m = \gamma k$ for any constant $0 < \gamma < 1$ in this case, as we do when $k \in \omega(\log n)$, or is there some threshold below which we simply cannot recoup most of the randomness of the un-exposed bits of the input? We state this question formally below.

Open Question 5.2. Does there exist, for every constant $0 < \gamma < 1$, a weakly static statistical (k,ε)-ERF from n bits to $m = \gamma k$ bits with k constant and $\varepsilon \leq 1/2$?

[1] In the literature, these are simply called "condensers".

Note that this question is actually an open one for *all* of the various statistical ERF's and RF's that we have discussed. In particular, if we replace "weakly static statistical ERF" with "statistical RF", we arrive at the simple and fundamental question of whether there is a limit to how well we can deterministically extract randomness from oblivious bit-fixing sources with a constant number of random bits.

There are two reasons that we might think that there *is* a trade-off in the statistical setting. First of all, it seems difficult to believe that for constant k we can really extract any fixed fraction of the randomness from an (n, k) oblivious bit-fixing source no matter how much interference we throw in by making n arbitrarily large.

The second reason first requires a bit of background: Kamp and Zuckerman [12] consider a generalization of the problem of randomness extraction from oblivious bit-fixing sources to sources called oblivious *symbol*-fixing sources. Symbol-fixing sources are a generalization of bit-fixing sources that, instead of taking values in $\{0, 1\}^n$, take values in $[d]^n$ for some fixed $d \geq 2$. In the case of oblivious symbol-fixing sources with $d > 2$, a deterministic extractor can be constructed by using the symbols of the input to take a random walk on a d-regular *expander* graph.[2] Since random walks on expander graphs converge to the uniform distribution very quickly, this method yields extractors that do achieve the goal of extracting γk bits for any constant $0 < \gamma < 1$ even when k is constant.

So what goes wrong when $d = 2$? The problem is that 2-regular expanders do not exist. Kamp and Zuckerman [12] give a method of converting a bit-fixing source into an approximation of a symbol-fixing source with $d > 2$ and then using the expander graph method described above. This results in the extractor of Theorem 2.17 (discussed in Section 4.2.3), but it requires a lot of randomness (specifically, $k \gg \sqrt{n}$). The alternative is to take a walk on the only reasonable 2-regular graph possible: a cycle. Doing so produces the previously mentioned extractor (and weakly static ERF) of Theorem 3.20, which does work for all k but has output length only $O(\log k)$. Here lies the second reason that the answer to Open Question 5.2 may be negative: perhaps the basic fact that expanders of degree two do not exist is actually related to the fundamental limitations of weakly static ERF's.

The Search for an Answer: Attempting to Protect Fewer Bits of the Input

Answering Open Question 5.2 would provide insight into a basic fact about extraction from oblivious bit-fixing sources, and either of the two types of resilience amplification that we have outlined could help us settle it. We could apply a method of increasing output length to the ERF that works for all k but has $m = (\log k)/4$ (Theorem 3.20) until it has a better output length; or we could apply a method of decreasing k/n to the ERF that has $m = \gamma k$ for any constant $0 < \gamma < 1$ but requires $k \in \omega(\log n)$ (Theorem 3.18) until it requires very few protected bits. We will focus on finding a way to decrease the number of input bits requiring protection.

[2] Recall that an expander graph is a graph that is efficiently constructible yet highly connected despite having few edges.

Since ERF's are analogous in many ways to randomness extractors, we ask whether condensers for seeded extractors, which have become a key element of many extractor constructions, have analogues for exposure-resilient functions. If such analogues can be constructed, they could settle Open Question 5.2 in the affirmative; on the other hand, if they do not exist, we can interpret this as perhaps pointing to the conclusion that the answer to Open Question 5.2 is a negative one.

In the next section, we will pursue this idea by adapting the definition of condensers for seeded extractors to the setting of weakly static ERF's, calling the resulting objects *condensers for WSERF's*. We will then proceed to show in the rest of the chapter that condensers for WSERF's do not exist for any non-trivial settings of parameters, thus ruling out this natural approach to amplifying exposure-resilience and suggesting that the answer to Open Question 5.2 may indeed by negative.

All of our results will also apply to the setting of perfect RF's and ERF's since the results all hold for $\varepsilon = 0$.

5.2 The Definition of Lossless Condensers for WSERF's

Having seen what developing a theory of condensers for WSERF's can offer us, we now begin our investigation in depth. Taking our cue from the salient role of lossless condensers in the randomness extractor literature, we will focus on *lossless* condensers for WSERF's.

Let us try to deduce properties that would be natural to require of a lossless condenser for WSERF's $C\colon \{0,1\}^{n'} \to \{0,1\}^n$. To simplify our discussion as we do this, we will assume that $n' = n+1$. What properties ensure that, for any statistical (k,ε)-WSERF $f\colon \{0,1\}^n \to \{0,1\}^m$, the function $f \circ C\colon \{0,1\}^{n+1} \to \{0,1\}^m$ will be a statistical $(k, \varepsilon+\delta)$-WSERF?

We turn again to our alternate characterization from Section 3.2. We know that a statistical (k,ε)-WSERF f extracts on average from sets of sources of the form $\mathcal{F}_L = \{\langle L^{a,n}\rangle : a \in \{0,1\}^{n-k}\}$ for all $L \in \left\{{n \atop n-k}\right\}$. We also see that for $f \circ C$ to be a weakly static ERF it must extract from sets of sources of the form $\mathcal{G}_L = \{\langle L^{a,n+1}\rangle : a \in \{0,1\}^{n+1-k}\}$ for all $L \in \left\{{n+1 \atop n+1-k}\right\}$. Imagining for a second that the error parameter δ of C is 0, a natural way to build a function C will suffice is to ask that for every $L \in \left\{{n+1 \atop n+1-k}\right\}$ there exist some $S \in \left\{{n \atop n-k}\right\}$ such that:

- C maps each of the 2^{n+1-k} sources in $\mathcal{G}_L$ bijectively onto some source in $\mathcal{F}_S$.
- Each of the 2^{n-k} sources in $\mathcal{F}_S$ is mapped to in this way by exactly two sources in $\mathcal{G}_L$.

If C satisfies these conditions then we can easily deduce that $f \circ C$ is a weakly static statistical (k,ε)-ERF because an expectation over a source drawn uniformly at random from $\mathcal{G}_L$ will, after applying C, become an expectation over a source drawn uniformly at random from $\mathcal{F}_S$.

Of course, we do not actually want to limit ourselves to $\delta = 0$, so let us add that instead of mapping each source from $\mathcal{G}_L$ bijectively onto a source in $\mathcal{F}_S$, C need only map the former to within some statistical distance δ of the latter. These properties yield the definition below (generalized to input values beyond $n+1$) of lossless condensers for WSERF's.

Definition 5.3 (Lossless Condenser for WSERF's)**.** A function $C\colon \{0,1\}^{n'} \to \{0,1\}^n$ (where $n' > n$) is a *lossless* (k,δ)*-condenser for WSERF's* if and only if there exists a pair of functions $\varphi\colon \left\{ {n' \atop n'-k} \right\} \to \left\{ {n \atop n-k} \right\}$ and $\Psi\colon \left\{ {n' \atop n'-k} \right\} \times \{0,1\}^{n'-k} \to \{0,1\}^{n-k}$ with the following properties:

1. For every $L \in \left\{ {n' \atop n'-k} \right\}$ and every $a \in \{0,1\}^{n'-k}$, we have $C(\langle L^{a,n'} \rangle) \cong_\delta \langle \varphi(L)^{\Psi(L,a),n} \rangle$.
2. For every $L \in \left\{ {n' \atop n'-k} \right\}$, the function $\Psi(L,-)$ is regular in its second parameter.[3]

The function φ is called the *set function* of C and the function Ψ is called its *bit function*.

Our discussion above proves the following proposition, which formalizes the capability of lossless condensers for WSERF's to amplify resilience.

Proposition 5.4. *Let* $f\colon \{0,1\}^n \to \{0,1\}^m$ *be a statistical (resp. computational)* (k,ε)*-WSERF and let* $C\colon \{0,1\}^{n'} \to \{0,1\}^n$ *be a lossless* (k,δ)*-condenser for WSERF's. Then the function* $f \circ C\colon \{0,1\}^{n'} \to \{0,1\}^m$ *is a statistical (resp. computational)* $(k,\varepsilon+\delta)$*-WSERF.*[4]

Definition 5.3 would be simpler if we knew that the functions φ and Ψ are always unique. Proposition 5.5 below shows that since oblivious bit-fixing sources are statistically far apart, this is indeed the case as long as $\delta < 1/4$. In the rest of this chapter we will only be dealing with values of δ less than $1/4$ and so we will treat the set functions and bit functions of our condensers as unique without explicitly citing this proposition.

Proposition 5.5. *Let* $\langle W \rangle = \langle S^{w,n} \rangle$ *and* $\langle X \rangle = \langle T^{x,n} \rangle$ *be two distinct* (n,k) *oblivious bit-fixing sources. Let* $\langle D \rangle$ *be any distribution over* $\{0,1\}^n$*. Then for* $\delta < 1/4$*, we cannot have both* $\langle D \rangle \cong_\delta \langle W \rangle$ *and* $\langle D \rangle \cong_\delta \langle X \rangle$*.*

Proof. Proposition B.2 (see Appendix B) tells us that $\Delta(\langle W \rangle, \langle X \rangle) \geq 1/2$. Supposing that $\langle D \rangle$ is δ-close to both $\langle W \rangle$ and $\langle X \rangle$, we can use the triangle inequality for Δ (Lemma 2.6 from the our preliminaries) to see that $2\delta \geq 1/2$, implying that $\delta \geq 1/4$. □

The definition of lossless condensers for WSERF's has analogues for weakly adaptive statistical ERF's as well as statistical RF's. However, we do not state them here since the results of the rest of this chapter apply only to lossless condensers for WSERF's. Additionally, in the remainder of this chapter we will refer to lossless condensers for WSERF simply as "lossless condensers" except in statements of lemmas and theorems.

[3] Recall that a regular function is a function that maps the same number of elements of its domain to each element of its target space.

[4] Note that this proposition encompasses the perfect case as well since we can set $\varepsilon = \delta = 0$.

5.3 Impossibility in the Perfect Setting

For a lossless (k,δ)-condenser $C\colon \{0,1\}^{n'} \to \{0,1\}^n$ to successfully amplify the resilience of a *perfect* k-RF, we need $\delta = 0$. However, it is simple to show that this is impossible to have for all $0 < k < n < n'$.

Proposition 5.6. *The following statements hold for every $k \geq 2$:*

1. *There is no lossless $(k,0)$-condenser for WSERF's $C\colon \{0,1\}^{3k+1} \to \{0,1\}^k$.*
2. *There exists an $n \in \{k,\dots,3k\}$ such that there is no lossless $(k,0)$-condenser for WSERF's $C\colon \{0,1\}^{n+1} \to \{0,1\}^n$.*

Proof. Any lossless $(k,0)$-condenser $C\colon \{0,1\}^{3k+1} \to \{0,1\}^k$ is a perfect k-RF, and so its existence directly contradicts the previously discussed result of Chor et al. [23], which states that a perfect k-RF with more than one output bit must have k equal at least one-third of its input length. This proves the first statement.

As for the second statement: if for every $n \in \{k,\dots,3k\}$ there exists a function $C_n\colon \{0,1\}^{n+1} \to \{0,1\}^n$ that is a lossless $(k,0)$-condenser, then $C_k \circ \cdots \circ C_{3k}$ is a lossless $(k,0)$-condenser from $3k+1$ bits to k bits, which contradicts the first statement. □

This impossibility result only partially answers our question though, because for each k it only gives us *certain* settings of parameters for which it is impossible to have a lossless condenser. In the following section we prove a much stronger result for the statistical case that subsumes this one (by setting $\varepsilon = 0$): that there do not exist lossless condensers for *any* non-trivial settings of parameters.

5.4 Impossibility in the Statistical Setting

We now show that for δ less than some fixed constant there do not exist lossless (k,δ)-condensers from n' bits to n bits for any $0 < k < n < n'$. Before we begin the proof, though, let us sketch its strategy.

5.4.1 A Proof Sketch

The basic fact that we prove and then exploit is that, for sufficiently small δ, no function from $n+1$ bits to n bits can be both a lossless (n,δ)-condenser and a lossless $(n-1,\delta)$-condenser. We prove this in Lemma 5.11, but before doing so we show that we can take any given lossless (k,δ)-condenser $C\colon \{0,1\}^{n'} \to \{0,1\}^n$ and convert it into a function $C'\colon \{0,1\}^{k+1} \to \{0,1\}^k$ that is both a lossless (k,δ)-condenser and a lossless $(k-1,6\delta)$ condenser, giving us a contradiction.

We build C' in three stages (after reducing to the special case of $n' = n+1$ in Lemma 5.7).

1. Lemma 5.8: We prove that there exists an $(n+1, k+1)$-OBFS $\langle L^{a,n+1}\rangle$ that is mapped by C approximately to an (n,k)-OBFS $\langle S^{w,n}\rangle$. We will construct C' by restricting C to $L^{a,n+1}$ and use this result to show that we can effectively treat $S^{w,n}$ as the target space of C.
2. Lemma 5.9: We show that C maps every $(n+1,k)$-OBFS whose support is contained in $L^{a,n+1}$ approximately to $\langle S^{w,n}\rangle$. This is how we will know that the function C' is a lossless (k,δ)-condenser.
3. Lemma 5.10: We show that C maps every $(n+1, k-1)$-OBFS whose support is contained in $L^{a,n+1}$ approximately to an $(n, k-1)$-OBFS whose support is contained in $\langle S^{w,n}\rangle$. This allows us to say that C' is a lossless $(k-1, 6\delta)$-condenser.

By putting together all the steps above and applying Lemma 5.11, we arrive at our final result, which we state precisely in Theorem 5.12.

5.4.2 The Lemmas

We start by reducing our task to the special case of lossless condensers from $n+1$ bits to n bits.

Lemma 5.7. *If there exists a lossless (k,δ)-condenser for WSERF's $C\colon \{0,1\}^{n'} \to \{0,1\}^n$, then there exists a lossless (k,δ)-condenser for WSERF's $D\colon \{0,1\}^{n+1} \to \{0,1\}^n$.*

Proof. Define $D(x) = C(x \circ 0^{n'-(n+1)})$. Then any $(n+1,k)$-OBFS given to D will be passed to C as an (n',k)-OBFS and so D's output will have the desired properties. □

We now begin proving the three main lemmas outlined in our proof sketch. Our first step is to establish that there is $(n+1, k+1)$ oblivious bit-fixing source that is mapped by our lossless condenser C to within distance δ of an (n,k) oblivious bit-fixing source. In proving this lemma we make use of Lemma B.4 from Appendix B, which gives a lower bound on the size of the pre-images of sets that are known to be approximately mapped to by certain sets of oblivious bit-fixing sources. Lemma B.4 relies purely on basic properties of oblivious bit-fixing sources.

Lemma 5.8. *Let $C\colon \{0,1\}^{n+1} \to \{0,1\}^n$ be a lossless (k,δ)-condenser for WSERF's with $\delta < 1/4$ and $0 < k < n$. Then there exists an $(n+1, k+1)$ oblivious bit-fixing source $\langle L^{a,n+1}\rangle$ and an (n,k) oblivious bit-fixing source $\langle S^{w,n}\rangle$ such that $C(\langle L^{a,n+1}\rangle) \cong_\delta \langle S^{w,n}\rangle$.*

Proof. Let φ and Ψ be the set function and bit function of C respectively. Since $\left|\left\{ {n+1 \atop n+1-k} \right\}\right| = \binom{n+1}{k} > \binom{n}{k} = \left|\left\{ {n \atop n-k} \right\}\right|$, there exists some set $S \in \left\{ {n \atop n-k} \right\}$ and two sets $L_1, L_2 \in \left\{ {n+1 \atop n+1-k} \right\}$ such that $\varphi(L_1) = \varphi(L_2) = S$. Since $\Psi(L_i, -)$ is two-to-one, we know that for every $w \in \{0,1\}^{n-k}$ and every

$i \in \{1,2\}$, there exist two strings $a_i, b_i \in \{0,1\}^{n+1-k}$ such that $C(\langle L_i^{a_i,n+1}\rangle)$ and $C(\langle L_i^{b_i,n+1}\rangle)$ are both δ-close to $\langle S^{w,n}\rangle$.

We will show that there exists at least one w for which the strings a_1 and b_1 differ in only one bit. We will then define the OBFS $L^{a,n+1}$ to be the OBFS fixing the bits in positions at which a_1 and b_1 agree, which will give us the $(n+1, k+1)$-OBFS that we seek.

Suppose for contradiction that the strings a_1 and b_1 differ in at least two bits for every w. Then we can apply Lemma B.4 with $\langle S\rangle = \langle S^{w,n}\rangle$ to conclude that, for each w, the size of $C^{-1}(\langle S^{w,n}\rangle)$ is at least $2^{k+1} + 2^k - \delta 2^{k+2}$. Now we can sum over all $w \in \{0,1\}^{n-k}$ to get a lower bound on $\bigsqcup_w C^{-1}(S^{w,n}) = C^{-1}(\{0,1\}^n)$, which we know to be of size at most 2^{n+1} (the size of the domain). We do so below:

$$\begin{aligned}
2^{n+1} &\geq \sum_{w\in\{0,1\}^{n-k}} \left(2^{k+1} + 2^k - \delta 2^{k+2}\right) \\
&= 2^{n-k}\left(2^{k+1} + 2^k - \delta 2^{k+2}\right) \\
&= 2^{n+1} + 2^n - \delta 2^{n+2} \\
\Rightarrow \qquad \delta &\geq \frac{1}{4}
\end{aligned}$$

Of course, this is a contradiction since $\delta < 1/4$. Thus, we have some $w \in \{0,1\}^{n-k}$ and two strings (call them a_0 and b_0) in $\{0,1\}^{n+1-k}$, differing in exactly one position, such that $C(\langle L_1^{a_0,n+1}\rangle)$ and $C(\langle L_1^{b_0,n+1}\rangle)$ are both δ-close to $\langle S^{w,n}\rangle$. Define $L \subset L_1$ to be the set of positions at which the bits of a_0 and b_0 agree, and set a to be $[L_1^{a_0,n+1}]_L = [L_1^{b_0,n+1}]_L$. Note that $|L| = n-k$ and so $\langle L^{a,n+1}\rangle$ is an $(n+1, k+1)$ oblivious bit-fixing source. Furthermore, $C(\langle L^{a,n+1}\rangle)$ is equal to $\frac{1}{2}C(\langle L_1^{a_0,n+1}\rangle) + \frac{1}{2}C(\langle L_1^{b_0,n+1}\rangle)$, allowing us to apply Lemma 2.6 from the preliminaries to manipulate the statistical distance of interest into what we want.

$$\begin{aligned}
\Delta(C(\langle L^{a,n+1}\rangle), \langle S^{w,n}\rangle) &= \Delta\left(\frac{1}{2}C(\langle L_1^{a_0,n+1}\rangle) + \frac{1}{2}C(\langle L_1^{b_0,n+1}\rangle), \langle S^{w,n}\rangle\right) \\
&\leq \frac{1}{2}\Delta(C(\langle L_1^{a_0,n+1}\rangle), \langle S^{w,n}\rangle) + \frac{1}{2}\Delta(C(\langle L_1^{b_0,n+1}\rangle), \langle S^{w,n}\rangle) \\
&\leq \delta
\end{aligned}$$

This completes the proof. □

Having obtained the "special" pair of oblivious bit-fixing sources $\langle L^{a,n+1}\rangle$ and $\langle S^{w,n}\rangle$, we now need to say what happens to the $(n+1, k)$-OBFS's whose supports are contained in $L^{w,n+1}$. Namely, we need to show that they map more or less to $\langle S^{w,n}\rangle$. This is not too difficult, because they all must get mapped to OBFS's since C is a condenser, and if one of them is mapped to an OBFS that is not $\langle S^{w,n}\rangle$ it must get mapped *very* far from $\langle S^{w,n}\rangle$ since distinct OBFS's are far apart from each other. This then keeps $\langle L^{a,n}\rangle$ from being mapped to within distance δ of $\langle S^{w,n}\rangle$, producing a contradiction. The following lemma formalizes this argument.

Lemma 5.9. *Let $C\colon \{0,1\}^{n+1} \to \{0,1\}^n$ be a lossless (k,δ)-condenser for WSERF's with $\delta < 1/6$ and $0 < k < n$. Suppose that there is an $(n+1,k+1)$ oblivious bit-fixing source $\langle L^{a,n+1}\rangle$ and an (n,k) oblivious bit-fixing source $\langle S^{w,n}\rangle$ such that $C(\langle L^{a,n+1}\rangle) \cong_\delta \langle S^{w,n}\rangle$.*

Then for every $i \notin L$ and every $\beta \in \{0,1\}$, we have that C maps the $(n+1,k)$ oblivious bit-fixing source $\langle L^{a,n+1} \cap \{i\}^{\beta,n+1}\rangle$ to within distance δ of $\langle S^{w,n}\rangle$.

Proof. Let $M = L \cup \{i\}$ and let $b \in \{0,1\}^{n+1-k}$ be the unique string such that $M^{b,n+1} = L^{a,n+1} \cap \{i\}^{\beta,n+1}$. Suppose for contradiction that $C(\langle M^{b,n+1}\rangle)$ is δ-close to some *other* (n,k)-OBFS $\langle V\rangle$. We know from Proposition B.2 that $\Delta(\langle S^{w,n}\rangle, \langle V\rangle) \geq 1/2$, and since both distributions are uniform distributions on 2^k strings, this implies that their supports overlap in at most 2^{k-1} strings and so $|V - S^{w,n}| \geq 2^{k-1}$.

But since $C(\langle L^{a,n+1}\rangle) \cong_\delta C(\langle S^{w,n}\rangle)$ and $M^{b,n+1} \subset L^{a,n+1}$, we know that C sends at most $\delta 2^k$ of the strings in $M^{b,n+1}$ to $V - S^{w,n}$. This means that $\Delta(C(\langle M^{b,n+1}\rangle), \langle V\rangle)$ is at least $(2^{k-1} - 2\delta 2^k)/2^k$ (because a distinguisher could accept precisely the strings in $V - C(M^{b,n+1})$). On the other hand though, this distance must be at most δ. The resultant inequality, $\delta \geq 1/2 - 2\delta$ implies that $\delta \geq 1/6$, a contradiction. □

In the third of our three main lemmas, we say what happens to the $(n+1,k-1)$-OBFS's with supports contained in $L^{a,n+1}$: they get mapped approximately to $(n,k-1)$-OBFS's in the target space that are contained in $S^{w,n}$.

Lemma 5.10. *Let $C\colon \{0,1\}^{n+1} \to \{0,1\}^n$ be a lossless (k,δ)-condenser for WSERF's with $\delta < 1/4$ and $0 < k < n$. Suppose that there is an $(n+1,k+1)$ oblivious bit-fixing source $\langle L^{a,n+1}\rangle$ and an (n,k) oblivious bit-fixing source $\langle S^{w,n}\rangle$ such that $C(\langle L^{a,n+1}\rangle) \cong_\delta \langle S^{w,n}\rangle$.*

Then for every $i,j \notin L$ and every $\beta,\gamma \in \{0,1\}$, we have that C maps the $(n+1,k-1)$ oblivious bit-fixing source $\langle L^{a,n+1} \cap \{i,j\}^{\beta\gamma,n+1}\rangle$ to within distance 6δ of an $(n,k-1)$ oblivious bit-fixing source whose support is contained in $S^{w,n}$.

Proof. Without loss of generality we may take L and S to be $[n-k]$. Let a' be a with its final bit removed, and define the following sets:

$$\begin{aligned} M &:= L \cup \{i\} \\ N &:= L \cup \{i,j\} \backslash \{n-k\} \\ \hat{L} &:= L \cup \{i,j\} \end{aligned}$$

The source that we're interested in proving something about is $\langle \hat{L}^{a\beta\gamma,n+1}\rangle$, and we note that its support is equal to the intersection of the supports of $\langle M^{a\beta,n+1}\rangle$ and $\langle N^{a'\beta\gamma,n+1}\rangle$, both of which are (n,k)-OBFS's.

Since $M^{a\beta,n+1}$ and $N^{a'\beta\gamma,n+1}$ are $(n+1,k)$-OBFS's, we know by C's condenser property that each is mapped to within distance δ of some (n,k)-OBFS. Further, since $M^{a\beta,n+1} \subset L^{a,n+1}$, Lemma 5.9 tells us more specifically that $\langle M^{a\beta,n+1}\rangle$ is mapped to within distance δ of $\langle S^{w,n}\rangle$. Thus, we may write:

$$\begin{aligned} C(\langle M^{a\beta,n+1}\rangle) &\cong_\delta \langle S^{w,n}\rangle \\ C(\langle N^{a'\beta\gamma,n+1}\rangle) &\cong_\delta \langle T^{x,n}\rangle \end{aligned}$$

where $\langle T^{x,n}\rangle$ is some (n,k)-OBFS. We will prove our lemma by establishing the following escalating sequence of claims concerning $S^{w,n}$ and $T^{x,n}$, the last two of which directly imply the result.

1. C maps most of $\hat{L}^{a\beta\gamma,n+1}$ into $S^{w,n} \cap T^{x,n}$.
2. $\langle S^{w,n} \cap T^{x,n}\rangle$ is an (n,k')-OBFS with $k' < k$.
3. $\Delta(C(\langle \hat{L}^{a\beta\gamma,n+1}\rangle), \langle S^{w,n} \cap T^{x,n}\rangle) \leq 6\delta$
4. $k' = k-1$, making $\langle S^{w,n} \cap T^{x,n}\rangle$ our desired $(n,k-1)$-OBFS.

Justifications of each claim follow.

Claim 1: At least a $(1-4\delta)$ fraction of the strings in $\hat{L}^{a\beta\gamma,n+1}$ are mapped by C into $S^{w,n} \cap T^{x,n}$.
Since $\hat{L}^{a\beta\gamma,n+1} \subset M^{a\gamma,n+1}$ and $C(\langle M^{a\gamma,n+1}\rangle) \cong_\delta \langle S^{w,n}\rangle$, at most $\delta 2^k$ of the strings in $\hat{L}^{a\beta\gamma,n+1}$ can be carried outside of $S^{w,n}$ by C. By the same argument (using $N^{a'\beta\gamma,n+1}$), we know that at most $\delta 2^k$ of the strings in $\hat{L}^{a\beta\gamma,n+1}$ can be carried outside of $T^{x,n}$ by C. Therefore, at most $\delta 2^{k+1}$ strings in $\hat{L}^{a\beta\gamma,n+1}$ can be carried outside of $S^{w,n} \cap T^{x,n}$ by C, leaving at least $2^{k-1} - \delta 2^{k+1}$ strings, which is a $(1-4\delta)$ fraction of the 2^{k-1} strings in $\hat{L}^{a\beta\gamma,n+1}$, as desired.

Claim 2: $\langle S^{w,n} \cap T^{x,n}\rangle$ is a (n,k')-OBFS with $k' < k$.
By Claim 1, $S^{w,n} \cap T^{x,n}$ is not empty and so we can apply Proposition B.1 (which characterizes intersections of OBFS's) to see that $\langle S^{w,n} \cap T^{x,n}\rangle$ is an (n,k')-OBFS with $k' \leq k$ and with $k = k'$ if and only if $S^{w,n} = T^{x,n}$. The following argument shows that $S^{w,n} \neq T^{x,n}$.

Recall that since $\Psi(M,-)$ is two-to-one, the *only* strings z for which $\langle M^{z,n+1}\rangle$ is carried by C to within distance δ of $S^{w,n}$ are $a\beta$ and $a\overline{\beta}$ (since we already know by Lemma 5.9 that $a\beta$ and $a\overline{\beta}$ have this property). Since $\varphi(M) = S$, this means that if we set z to anything else then $C(\langle M^{z,n+1}\rangle) \cong_\delta \langle S^{w',n}\rangle$ for some $S^{w',n}$ that is *disjoint* from $S^{w,n}$. Applying this to $z = \overline{a}\beta$, where $\overline{a}$ denotes a with its final bit flipped, we see that $C(\langle M^{\overline{a}\beta,n+1}\rangle) \cong_\delta \langle S^{w',n+1}\rangle$ for some (n,k)-OBFS $\langle S^{w',n}\rangle$ whose support is disjoint from $S^{w,n}$.

Now assume for contradiction that $S^{w,n} = T^{x,n}$. The size of the intersection of $M^{\overline{a}\beta,n+1}$ with $N^{a'\beta\gamma,n+1}$ is 2^{k-1}, and since $C(\langle N^{a'\beta\gamma,n+1}\rangle) \cong_\delta \langle T^{x,n}\rangle = \langle S^{w,n}\rangle$, at least $2^{k-1} - \delta 2^k$ of the strings in that intersection (and therefore in $M^{\overline{a}\beta,n+1}$) must be mapped into $S^{w,n}$. But since $S^{w,n}$ and $S^{w',n}$ are disjoint, none of the strings mapped to $S^{w,n}$ are mapped to $S^{w',n}$. This implies that

$$\delta \geq \Delta(C(\langle M^{\overline{a}\beta}\rangle), \langle S^{w',n}\rangle) \geq 2^{k-1} - \delta 2^k$$

When solved for δ, the above equation gives $\delta \geq 2^{k-1}/(1+2^k)$ which is at least $1/3$ for $k \geq 1$, a contradiction.

Claim 3: $\Delta(C(\langle \hat{L}^{a\beta\gamma,n+1}\rangle), \langle S^{w,n} \cap T^{x,n}\rangle) \leq 6\delta$.
Let $Y = S^{w,n} \cap T^{x,n}$, the support of our candidate $(n, k-1)$-OBFS. We bound $\Delta(C(\langle \hat{L}^{a\beta\gamma,n+1}\rangle), \langle Y\rangle)$ by counting only the strings that are more likely to be drawn from $C(\langle \hat{L}^{a\beta\gamma,n+1}\rangle)$ than from $\langle Y\rangle$ (see Proposition 2.5 in the preliminaries). Recalling that $\langle Y\rangle$ is a flat distribution on $2^{k'}$ elements, we see that each such string $z \in \{0,1\}^n$ falls into precisely one of the following two categories:

1. $z \notin Y$, but it is mapped to by one or more strings in $\hat{L}^{a\beta\gamma,n+1}$.
2. $z \in Y$, and it is mapped by by more than $2^{k-1-k'}$ strings in $\hat{L}^{a\beta\gamma,n+1}$.

By Claim 1, we know that there are at most $(4\delta)2^{k-1} = \delta 2^{k+1}$ strings that map to strings z that fall into the first category, so we need to bound, for each z in the second category, the number of strings in $\hat{L}^{a\beta\gamma,n+1}$ beyond the first $2^{k-1-k'}$ that map to z. Let $\#_z$ be the number of such strings for each $z \in Y$. The key to bounding $\#_z$ is to realize that $\hat{L}^{a\beta\gamma,n+1} \subset M^{a\beta,n+1}$ means that, for each $z \in Y$, every string in $\hat{L}^{a\beta\gamma,n+1}$ beyond the *first* that is mapped to z by C contributes 2^{-k} to the distance between $C(\langle M^{a\beta,n+1}\rangle)$ and $\langle S^{w,n}\rangle$, which is at most δ. Since k' is at most $k-1$, $2^{k-1-k'} \geq 1$ and so each string counted by $\#_z$ is at least the second string to be mapped to z, meaning that *every* string counted by $\#_z$ contributes 2^{-k} to $\Delta(C(\langle M^{a\beta,n+1}\rangle), \langle S^{w,n}\rangle) \leq \delta$. Therefore, if we let $\# = \sum_z \#_z$ then $\#/2^k \leq \delta$, implying that $\# \leq \delta 2^k$.

Each of the strings in either the first of second categories above contributes $2^{-(k-1)}$ to the distance $\Delta(C(\langle \hat{L}^{a\beta\gamma}\rangle), \langle Y\rangle)$, so our above argument shows that that distance is at most

$$\frac{\delta 2^{k+1} + \delta 2^k}{2^{k-1}} = 6\delta$$

Claim 4: $k' = k-1$.
Proposition B.1, which we used in the proof of Claim 2 to deduce that $\langle S^{w,n} \cap T^{x,n}\rangle$ is an (n, k')-OBFS with $k' < k$, states that k' is determined only by S and T, which are both determined only by L (and N, but N is determined by L as well). Thus, k' is independent of the settings of the bits β and γ, and so if we assume for contradiction that $k' < k-1$, we have that at least a $(1-6\delta)$ fraction of the set $\hat{L}^{a\beta\gamma,n+1}$ is mapped by C into a set of size at most 2^{k-2} for *all* values of β and γ. Since $\cup_{\gamma\in\{0,1\}} \hat{L}^{a\beta\gamma,n+1} = M^{a\beta,n+1}$, this means that at least a $(1-6\delta)$ fraction of $M^{a\beta,n+1}$ is mapped into a set of size $2 \cdot 2^{k-2} = 2^{k-1}$. By accepting precisely the strings in this set, an adversary could distinguish between $C(\langle M^{a\beta,n+1}\rangle)$ and $\langle S^{w,n}\rangle$ with advantage at least $(1-6\delta) - 1/2 = 1/2 - 6\delta$ because $\langle S^{w,n}\rangle$ is a flat distribution. But $C(\langle M^{a\beta,n+1}\rangle) \cong_\delta \langle S^{w,n}\rangle$, implying that $1/2 - \delta \leq \delta$, which means that $\delta \geq 1/4$, a contradiction. □

In our final lemma, we prove the basic fact that gives us our ultimate contradiction: that a function from $n+1$ bits to n bits cannot be a lossless condenser both for $(n+1, n)$-OBFS's and for

$(n+1, n-1)$-OBFS's. The proof of this lemma is simple in part because we can take advantage of the ideas already developed in Lemma 5.8.

Lemma 5.11. *For every integer $n > 1$, any function $C\colon \{0,1\}^{n+1} \to \{0,1\}^n$ that is both a lossless (n, δ)-condenser for WSERF's and a lossless $(n-1, \delta)$-condenser for WSERF's must have $\delta \geq 1/4$.*

Proof. Suppose for contradiction that there exists a function C as defined in the statement of the lemma but with $\delta < 1/4$. If we consider C as a lossless $(n-1, \delta)$-condenser, Lemma 5.8 tells us that there exists an $(n+1, n)$-OBFS $\langle L^{a,n+1} \rangle$ and an $(n, n-1)$-OBFS $\langle S^{w,n} \rangle$ such that $C(\langle L^{a,n+1} \rangle) \cong_\delta \langle S^{w,n} \rangle$.

On the other hand though, C is a lossless (n, δ)-condenser, so $C(\langle L^{a,n+1} \rangle) \cong_\delta \langle U_n \rangle$. By the triangle inequality for statistical distance (Lemma 2.6), we then have that $\langle S^{w,n} \rangle \cong_{2\delta} \langle U_n \rangle$. But $\Delta(\langle S^{w,n} \rangle, \langle U_n \rangle) = 1/2$ since $|S^{w,n}| = 2^{n-1}$, so we have a contradiction and therefore C cannot exist. □

5.4.3 The Result

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

Theorem 5.12. *There do not exist lossless (k, δ)-condensers for WSERF's from n' bits to n bits for any $0 < k < n < n'$ and any $\delta < 1/24$.*

Proof. Suppose for contradiction that we have such a lossless condenser C. We may assume, by Lemma 5.7, that $n' = n+1$. By Lemma 5.8, we know that we have an $(n+1, k+1)$-OBFS $\langle L^{a,n+1} \rangle$ and an (n, k)-OBFS $\langle S^{w,n} \rangle$ such that $C(\langle L^{a,n+1} \rangle) \cong_\delta \langle S^{w,n} \rangle$. We now define a new function $C'\colon \{0,1\}^{k+1} \to \{0,1\}^k$ as follows: given an input $x \in \{0,1\}^{k+1}$, C' creates a string $y \in \{0,1\}^{n+1}$ by setting $[y]_L = a$ and $[y]_{\overline{L}} = x$. C' then returns $[C(y)]_{\overline{S}}$ if $C(y) \in S^{w,n}$, and 0^k otherwise.

Lemma 5.9 and Lemma 5.10 show that if C^R is the restriction of C to $L^{a,n}$, C^R is a lossless (k, δ)-condenser and a lossless $(k-1, 6\delta)$-condenser respectively. But they tell us more than this: they also tell us that every $(k+1, k)$-OBFS (resp. $(k+1, k-1)$-OBFS) is mapped by C^R to within distance δ (resp. 6δ) of an OBFS whose support is *contained* in $S^{w,n}$. This means that modifying C^R by taking all the strings mapped outside of $S^{w,n}$ by C^R and re-mapping them anywhere within $S^{w,n}$ will not hurt C^R's performance as a lossless (k, δ)-condenser or as a lossless $(k-1, 6\delta)$-condenser.

However, this modification, together with removal of the (fixed) bits with positions in S, was exactly how we created C'. We therefore have that C' is a lossless (k, δ)-condenser and a lossless $(k-1, 6\delta)$-condenser; Lemma 5.11 then implies that $6\delta \geq 1/4$, a contradiction. □

5.5 References

All the results proven in this chapter and in Appendix B (which this chapter cites) are original.

Chapter 6

Further Study

As we have seen, resilient and exposure-resilient functions are both satisfying as objects of study and important as practical tools. A considerable amount is known about their properties and relationships, but there is much that is not yet understood both in terms of structural relationships and explicit constructions.

For example, in the statistical setting there are currently no useful explicit constructions of strong ERF's that are not also almost-perfect RF's. The same is true for the weakly adaptive ERF's: there are no non-trivial constructions of weakly adaptive ERF's that are not also RF's. This means that, at least for some settings of parameters, we may not have optimal constructions because of the separations proven in Chapter 4 which show that, for instance, the almost-perfect RF property is strictly stronger than the strongly adaptive ERF property. Relatedly, it would be nice to have a better understanding of exactly how pronounced the separations are between the different classes of RF's and ERF's. The separations that we proved in Chapter 4 are not necessarily as strong as possible, so in some sense our picture of the RF and ERF zoo is still incomplete.

It would also help to have better bounds on the parameters of the different types of functions, especially weakly static statistical ERF's (the weakest object in the statistical setting) and almost-perfect RF's (the strongest object). Such bounds would help us see how different the various RF's and ERF's really are from each other; more importantly though, they would also give us a better understanding of the intrinsic limits of extraction from oblivious bit-fixing sources. Recall that in Chapter 5 we posed an open question along these lines, asking whether there is a fundamental difference in terms of achievable output size between extraction from oblivious bit-fixing sources with constant entropy and oblivious bit-fixing sources with polylogarithmic entropy. To begin answering this question, we showed that lossless condensers for weakly static ERF's cannot exist for any non-trivial setting of parameters, thus eliminating a natural way of attempting to close the gap in output length between the constant and non-constant entropy cases. However, proving that such a gap really exists and making precise its extent remains an open question.

One possible approach to this challenge that continues along the lines we developed in Chapter 5 is to prove our impossibility result about lossless condensers for weakly static ERF's in greater generality. This could be done by ruling out the existence of *all* condensers for weakly static ERF's rather than just those condensers that suffer no entropy loss during the condensing process. However, a more general and much more powerful way to proceed would be to establish the impossibility of constructing *any* function C from n' bits to n bits that when composed with k-ERF's on n bits, gives k-ERF's on n' bits.

Lastly, since condensers for weakly static ERF's are new, their analogues for the cases of the other classes of RF's and ERF's have not been studied, and it is not obvious whether they exist or not. It would be interesting to find out whether, for instance, we can condense oblivious bit-fixing sources for the purposes of regular randomness extraction from oblivious bit-fixing sources even though we cannot do so for exposure-resilient extraction.

Appendix A

Error-Correcting Codes

An error-correcting code is an object designed to overcome the following difficulty: suppose Alice and Bob are communicating over a faulty channel and Alice wants a way to encode a message x such that if she sends the encoded message $E(x)$ over the channel then Bob will be able to retrieve x even if it is somehow corrupted before it reaches him (because of hardware failure, an adversary, or anything else). Naturally, analysis of such a problem requires a more formal way of modeling the faultiness of the channel and there are always different ways of doing this. However, we will only make an assumption about the worst-case behavior of the channel; that is, we will only assume that there exists some integer d such that strictly fewer than d of the bits of $E(x)$ are modified when it is sent over the channel. Under this assumption, we can then easily define an object that serves our purpose, and we do so below. Note that we have m as the input length and n as the output length here in order to keep the parametrization consistent with that of the discussion in Section 3.4.1.

Definition A.1 (Error-Correcting Code[1])**.** A function $E\colon \{0,1\}^m \to \{0,1\}^n$ is an *error-correcting code with minimum distance* d if and only if, for every pair of distinct strings $x, y \in \{0,1\}^m$ we have that $E(x)$ and $E(y)$ differ in at least d of their bits. Given such a an error-correcting code $E\colon \{0,1\}^m \to \{0,1\}^n$, we define its *rate* to be m/n.

It is easy to see that a function E will solve Alice and Bob's problem without fail if and only if it is an error-correcting code with minimum distance $2d$. Note that we have not guaranteed anything about the efficiency of the encoding and decoding processes and, further, that an efficient encoding is not necessarily efficiently decodable. This is of minor concern in this work since we deal only with codes that are linear maps from $\mathbb{F}_2^m$ to $\mathbb{F}_2^n$ and we are not interested in decoding our codes, but rather only want to use them as perfect RF's.

When designing error-correcting codes, we would of course like to maximize d (this corresponds to minimizing k when we treat E as a perfect RF). However, we would also like to have the rate of

[1] Typically, such codes are defined for alphabets more general than $\{0,1\}$, but in the interest of simplicity of presentation we leave this out here.

the code be as close to 1 as possible (this corresponds to getting as many output bits as possible from the perfect RF). We now prove some standard bounds on these parameters.

Proposition A.2 (Singleton Bound)**.** *Let $E\colon \{0,1\}^m \to \{0,1\}^n$ be an error-correcting code with minimum distance d. Then $m \leq n - d + 1$.*

Proof. Consider the set $C = E(\{0,1\}^m)$. Each element in C is distinct and, since every two elements differ in at least d places, we can delete the first $d-1$ bits of each string in C to obtain a new set with the same size. Since these new strings are all $n-(d-1)$ bits long, we have $|C| \leq 2^{n-d+1}$. On the other hand, $|C| = 2^m$, so we are done. □

The second bound we present is the one that shows us the limitations of linear perfect RF's.

Proposition A.3. *Let $E\colon \{0,1\}^m \to \{0,1\}^n$ be an error-correcting code with minimum distance d and with $m > \log n + 1$. Then $d < \lceil n/2 \rceil$.*

Proof. Suppose for contradiction that $d \geq \lceil n/2 \rceil$ and let $C = E(\{0,1\}^m)$ as before. Replace all the 0's in the elements of C with -1's and consider any two elements x and y of C as vectors in $\mathbb{R}^n$. Since x and y differ in at least $\lceil n/2 \rceil$ components, their dot product $x \cdot y$ contains at least $\lceil n/2 \rceil$ minus ones, and so we have $x \cdot y \leq 0$. We will prove, by induction on n, that because of this property C can contain no more than $2n$ vectors.

Let $A(n)$ denote the maximal size of any set $S \subset \mathbb{R}^n$ with the property that $x \cdot y \leq 0$ for all $x, y \in S$. The base case ($n = 1$) is clear, so $A(1) = 2$. Now let us consider $S \subset \mathbb{R}^n$ with this property. Without loss of generality (because we can change basis), the first vector in S is $(-1, 0, \ldots, 0)$. This implies that all the other vectors in S have a non-negative first component. If the vector $(1, 0, \ldots, 0)$ is also in S then all the other vectors in S have to lie in the $(n-1)$-dimensional subspace normal to $(1, 0, \ldots, 0)$ so we can apply the inductive hypothesis to see that $A(n) = 2 + A(n-1)$. If $(1, 0, \ldots, 0)$ is not in S then we can project every vector besides $(-1, 0, \ldots, 0)$ onto the subspace orthogonal to $(-1, 0, \ldots, 0)$. Since all the projected vectors have non-negative first components, this projection will only decrease their dot products with each other and so we can apply the inductive hypothesis to see that there are at most $A(n-1)$ of them. Since in this case we conclude that S can have at most $1 + A(n-1)$ vectors, we see that in general $A(n) \leq 2 + A(n-1)$. Of course, this recurrence is equivalent to $A(n) \leq 2n$, as desired.

Since $2^m = |C| \leq 2n$, we have that $m \leq \log 2n = \log n + 1$, a contradiction. □

Appendix B

Basic Facts about Oblivious Bit-Fixing Sources

We present here some basic properties of oblivious bit-fixing sources, their intersections with each other, and the statistical distance between them.

Our first proposition tells us what the intersection of two bit-fixing sources looks like.

Proposition B.1. *Let $\langle A \rangle = \langle L^{a,n} \rangle$ and $\langle B \rangle = \langle M^{b,n} \rangle$ be two distinct (n, k) oblivious bit-fixing sources. Then $A \cap B$ is either the empty set, or it is the support of an (n, k') oblivious bit-fixing source with $k' = n - |L \cup M| < k$.*

Proof. If the fixed bits of A and those of B disagree in any position $i \in L \cap M$, then the two sources are disjoint and the result is achieved, so we may assume that $[A]_{L\cap M} = [B]_{L\cap M}$. Call this string a. Any string $w \in A \cap B$, must have the bits in $L \cap M$ fixed to a. However, it must also have the bits in $L - M$ fixed to $[A]_{L-M}$ and the bits in $M - L$ fixed to $[B]_{M-L}$, and the rest of its bits can take on any value. Thus, $A \cap B$ is the support of an oblivious bit-fixing source on $\{0, 1\}^n$ with $|L \cup M|$ fixed bits. Further, if $L = M$ then A would be identical to B, so $|L \cup M| > n - k$ and consequently $k' < k$. □

We can apply Proposition B.1 to find the minimum statistical distance between any two (n, k)-OBFS's.

Proposition B.2. *Let $\langle A \rangle = \langle L^{a,n} \rangle$ and $\langle B \rangle = \langle M^{b,n} \rangle$ be two distinct (n, k) oblivious bit-fixing sources. Then $\Delta(\langle A \rangle, \langle B \rangle) \geq 1/2$*

Proof. If $L = M$ then the sources must have disjoint support and we are done, so we may assume that $|L \cup M| > n - k$. We know from Proposition B.1 that the intersection between the supports of these two sources has size at most 2^{k-1}, so if we choose a string from $\langle L^{a,n} \rangle$ we will also land in $\langle M^{b,n} \rangle$ at most $2^{k-1}/2^k = 1/2$ of the time, which implies the result. □

Our third proposition is slightly more involved. It essentially states that if we have two sources with the same set of fixed bits but whose actual fixed bit values differ in more than one place, then any other source will not have too much overlap with the union of the former two sources.

Proposition B.3. *Let $A = L^{a,n}$ and $B = L^{b,n}$ be the supports of two distinct (n, k) oblivious bit-fixing sources. If there is a third (n, k) oblivious bit-fixing source $\langle C\rangle$ distinct from $\langle A\rangle$ and $\langle B\rangle$ whose support intersects $A \cup B$ on more than half of its 2^k strings, then a and b agree in exactly $n - k - 1$ of their bits.*

Proof. Let the support of $\langle C\rangle$ be $C = M^{c,n}$. We first claim that for every $i \in M \cap L$, we must have $[L^{a,n}]_i = [L^{b,n}]_i$. To see this, assume the contrary: then one of the two of $[L^{a,n}]_i$ and $[L^{b,n}]_i$ (without loss of generality, say it's $[L^{a,n}]_i$) disagrees with $[M^{c,n}]_i$, so $C \cap A$ would be empty, meaning that $C \cap (A \cup B) = C \cap B$. Since $C \cap B$ is the support of an OBFS, it's size is a power of 2 and so it must be 2^k, meaning that $C = B$, a contradiction.

Having proven this claim, let us now show how it implies the result. As Proposition B.1 tells us, $C \cap A$ and $C \cap B$ are (n, k')-OBFS's with $k' = n - |L \cup M|$. And since $|C \cap (A \cup B)| > 2^{k-1}$, then without loss of generality $|C \cap A| > 2^{k-2}$, implying that $k' > k - 2$, which gives that $|L \cup M| < n - k + 2$. Because both L and M are of size $n - k$, we then have that $|L \cap M| > n - k - 2$. Since a and b agree at all the positions in $M \cap L$, they must agree in at least $n - k - 1$ positions. But they cannot agree in $n - k$ positions (or else they would be identical), so we have proved the proposition. □

The last statement that we prove here is a lemma that we use in our analysis of lossless condensers for WSERF's. It uses Proposition B.3 to give a lower bound on the size of the pre-image of a set that is mapped to by oblivious bit-fixing sources.

Lemma B.4. *Suppose that a function $C\colon \{0, 1\}^{n+1} \to \{0, 1\}^n$ maps each of the four $(n + 1, k)$ oblivious bit-fixing sources $\langle L_1^{a_1,n+1}\rangle$, $\langle L_1^{b_1,n+1}\rangle$, $\langle L_2^{a_2,n+1}\rangle$, and $\langle L_2^{b_2,n+1}\rangle$ to within statistical distance δ of some flat distribution $\langle S\rangle$, and that a_1 and b_1 differ in more than one bit. Then $|C^{-1}(S)| \geq 2^{k+1} + 2^k - \delta 2^{k+2}$.*

Proof. We know that $C^{-1}(S)$ contains at least $(1-\delta)2^k$ strings from each of the four OBFS's by the fact that it carries each to within δ of $\langle S\rangle$. However, we do not know to what extent these sources overlap. We know that $\langle L_1^{a_1,n+1}\rangle$ and $\langle L_1^{b_1,n+1}\rangle$ are disjoint, so we can count at least $2(1 - \delta)2^k$ strings in $C^{-1}(S)$. But what about $\langle L_2^{a_2,n+1}\rangle$ and $\langle L_2^{b_2,n+1}\rangle$? This is where Proposition B.3 comes in: since a_1 and b_1 differ in at least two places, we can use the proposition to conclude that at most half of each of $L_2^{a_2,n+1}$ and $L_2^{b_2,n+1}$ intersects the union $L_1^{a_1,n+1} \bigsqcup L_2^{b_1,n+1}$. Together with the fact that C carries each of these two sources to within δ of $\langle S\rangle$, we know that at most $\delta 2^k$ of the strings in $L_2^{a_2,n+1} - B_1$ are carried into S and that the same is true for $L_2^{b_2,n+1}$. We now have counted at least $2(1 - \delta)2^k + 2(2^{k-1} - \delta 2^k) = 2^{k+1} + 2^k - \delta 2^{k+2}$ strings in $C^{-1}(S)$, as desired. □

Bibliography

[1] P. Olofsson. *Probabilities: The Little Numbers that Rule Our Lives*, page 253. John Wiley and Sons, 2007.

[2] R.W. Clark. *The Man Who Broke Purple*. Weidenfeld and Nicolson, 1977.

[3] G. Marsaglia. Random numbers fall mainly in the planes. In *Proc. National Academy of Sciences*, volume 61, pages 25–28, 1968.

[4] Randu, March 2009. URL http://en.wikipedia.org/wiki/RANDU.

[5] M. Bellare and S. Miner. A forward-secure digital signature scheme. In *Proceedings of Cryptography*, pages 431–448, 1999.

[6] N. Someren. How not to authenticate code. crypto 1998 rump session, 1998. Santa Barbara.

[7] A. Shamir and N. Someren. Playing "hide and seek" with stored keys. In *Proceedings of Financial Cryptography*, 1999.

[8] A. Dornan. New viruses search for strong encryption keys. In *PlanetIT Systems Management News*, March 1999.

[9] R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai. Exposure-resilient functions and all-or-nothing transforms. In *Advances in Cryptography – EUROCRYPT 2000*, volume 1807/2000, pages 453–469, 2000. URL http://theory.csail.mit.edu/~cis/pubs/yevgen/aont.ps.

[10] K. Kurosawa, T. Johansson, and D. Stinson. Almost k-wise independent sample spaces and their cryptologic applications. In *Proceedings of EuroCrypt*, pages 409–421, 1997.

[11] A. Gabizon, R. Raz, and R. Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. In *Proc. 45th IEEE Symposium on Foundations of Computer Science*, pages 394–403, 2004. URL http://eccc.uni-trier.de/eccc-reports/2005/TR05-109/Paper.pdf.

[12] J. Kamp and D. Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In *FOCS 2003*, pages 92–101, 2003. URL http://www.cs.utexas.edu/users/diz/pubs/erf.pdf.

[13] N. Nisan. Extracting randomness: How and why: A survey. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 44 – 58, 1996.

[14] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. In *Journal of Computer and System Sciences*, volume 58, pages 148 – 173, 1999.

[15] S. Vadhan. Randomness extractors and their many guises. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 9 – 12, 2002.

[16] R. Shaltiel. Recent developments in explicit constructions of extractors. In *Bulletin of the EATCS*, volume 77, pages 67 – 95, 2002.

[17] V. Guruswami, C. Umans, and S. Vadhan. Unbalanced expanders and randomness extractors from parvareshvardy codes. In *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity (CCC 07)*, pages 96–108, June 2007. URL http://sosp16.cs.washington.edu/homes/venkat/pubs/papers/PV-condenser.pdf.

[18] J. Hastad, R. Impagliazzo, L.A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:12–24, 1999. URL http://www.icsi.berkeley.edu/~luby/PAPERS/hill.ps.

[19] O. Goldreich. Pseudorandom generators: A primer. 2008. URL http://www.wisdom.weizmann.ac.il/~oded/PS/prg08.pdf.

[20] S. Vadhan. Lecture notes for computer science 225: Pseudorandomness. 2007. URL http://www.eecs.harvard.edu/~salil/cs225/lecnotes/list.htm.

[21] Y. Dodis, A. Sahai, and A. Smith. On perfect and adaptive security in exposure-resilient cryptography. In *Advances in Cryptography – EUROCRYPT 2001*, volume 2045, pages 301–324, 2001. URL http://people.csail.mit.edu/asmith/PS/DSS-adaptive.ps.

[22] V. Byoko. On the security properties of oaep as an all-or-nothing transform. In *Proc. of Crypto.*, pages 503–518, 1999.

[23] B. Chor, O. Goldreich, J. Hastad, J. Freidmann, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In *Proc. 26th IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985. URL http://www.wisdom.weizmann.ac.il/~oded/PS/p3.ps.

[24] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. In *Proc. 38th IEEE Symposium on Foundations of Computer Science*, pages 585–594, 1997. URL http://eprints.kfupm.edu.sa/71975/1/71975.pdf.

[25] P. Diaconis. Group representations in probability and statistics. In *Lecture Notes–Monograph Series 11, Institute of Mathematical Statistics*, 1988. Hayward, CA.

[26] A. Lubotzky, R. Philips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, (8):353–398, 1988.

[27] O. Goldreich. A note on computational indistinguishability. In *IPL*, pages 277–281, 1990. URL http://www.wisdom.weizmann.ac.il/~oded/PS/iplnote.ps.

[28] C.H. Bennett, G. Brassard, and J.M. Robert. Privacy amplification by public discussion. *SIAM J. Computing*, 17(2):210–299, April 1988.

[29] R. Rivest. All-or-nothing encryption and the package transform. In *Fast Software Excryption, LNCS*, volume 1267, pages 210–219, 1997.