Master Thesis

Concrete constructions of unbalanced bipartite expander graphs and generalized conductors

Author(s):
Werner, Rose-Line

Publication Date:
2008

Permanent Link:
https://doi.org/10.3929/ethz-a-005664665

Rights / License:
In Copyright - Non-Commercial Use Permitted
Concrete Constructions of Unbalanced Bipartite Expander Graphs and Generalized Conductors

Rose-Line Werner

Master's Thesis in Computer Science
March - September 2008

Advisors
Prof. Dr. Ueli Maurer
Stefano Tessaro
This thesis is submitted for partial fulfillment of the requirements for the degree of Master of Science ETH in Computer Science at ETH Zurich (Swiss Federal Institute of Technology Zurich).

Acknowledgments

First of all, I want to thank Stefano Tessaro for his great support during my master’s thesis. We had a lot of valuable and instructive discussions and he gave me insights on how to write a scientific text. Furthermore, I highly appreciate that he spend many hours of his valuable time to proofread my thesis.

Also, I would like to thank Professor Dr. Ueli Maurer for being the responsible professor at ETH for my master’s thesis and for his brilliant lectures about Cryptography and Information Theory, which aroused my interest in this topic. They were the main reasons why I decided to write my thesis in the group of theoretical computer science.

I dedicate this thesis to my family and my friends, who have always given me strong support during my studies at ETH Zurich.

Waeldi, September 3rd, 2008

Rose-Line Werner
Abstract

Bipartite expander graphs are bipartite graphs for which the left vertex set has a guaranteed expansion parameter to the right vertex set, or more formally, a bipartite graph $G = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$ has expansion parameter $\gamma$ if for every set $\mathcal{X} \subset \mathcal{V}_1$ (with bounded size) with its neighbors $\Gamma(\mathcal{X}) \subset \mathcal{V}_2$, we have $|\Gamma(\mathcal{X})| \geq \gamma \cdot |\mathcal{X}|$. We are interested in the unbalanced case, where the set $\mathcal{V}_1$ is much larger than $\mathcal{V}_2$. If the neighbors of each vertex in $\mathcal{V}_1$ can be efficiently computed, the expander graph is additionally called explicit.

In this thesis, we introduce a generalized notion of conductors, which are functions of the form $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ which take as first input a string with certain min-entropy and as second input some few truly random bits, and provide some entropy guarantees on the output. These generalized conductors are usually interpreted as explicit unbalanced bipartite expander graphs with stronger properties. In particular, we provide strong composition theorems for such conductors which allow us to obtain explicit unbalanced bipartite expander graphs with good expansion parameters and sufficiently small left-degree, and to study the concrete values of these parameters.

Explicit unbalanced bipartite expander graphs with good expansion parameters can be used in cryptographic schemes (like the domain extender of public random functions due to Maurer and Tessaro). In particular, a small left-degree of the expander graph is crucial for the efficiency of the protocols using such expander graphs. Therefore, we focus on finding a construction of an explicit expander graph with small left-degree. We show non-constructively that such expander graphs (as well as other types of conductors) with small left-degree and good expansion must exist and try to find an explicit construction of such a good expander graph. In particular, we analyze an expander graph construction which leads to a small left-degree of the expander graph in complexity-theoretic terms and we investigate the concrete value of this left-degree. We show that even though the left-degree is polynomial, the actual degree of the polynomial makes it not feasible to use the construction in practice.

This give us the motivation to analyze an unbalanced bipartite expander graph construction which is based on selecting (according to some rule) substrings of length $n$ as the neighbors of a string which has length multiple of $n$: Although this construction has a small left-degree and promises a good expansion of the left vertices, we show that it is impossible to construct an expander graph with good expansion on the basis of substring selection.
# Contents

1 Introduction  
   1.1 Unbalanced Bipartite Expander Graphs .......................... 1  
   1.2 Domain Extension for Public Random Functions .................. 1  
   1.3 Contributions .................................................. 4  

2 Preliminaries and Notation  
   2.1 Notation ......................................................... 5  
   2.2 Probability and Information Theory .............................. 5  
      2.2.1 Entropy and Min-Entropy ................................ 5  
      2.2.2 Distributions .............................................. 6  
   2.3 Estimations for the Binomial Coefficient ....................... 9  
   2.4 Graph Theory .................................................. 10  

3 Expanders and Conductors  
   3.1 Generalized Unbalanced Bipartite Expander Graphs ............... 13  
   3.2 Generalized Conductors ........................................ 15  
   3.3 Composition Theorems for Conductors ........................... 18  
      3.3.1 Conductor Cascading ..................................... 18  
      3.3.2 Conductor Concatenation .................................. 19  
      3.3.3 Constructing Somewhere-Conductors by Conductor Cascading 22  
   3.4 Constructing Expander Graphs From Conductors ................. 25  
      3.4.1 Transforming Conductors into an Expander Graph .......... 26  
      3.4.2 Transforming Somewhere Conductors into an Expander Graph 27  
   3.5 Probabilistic Existence Proofs ................................ 27  
      3.5.1 Existence of Injective (Lossless) Conductor .............. 29  
      3.5.2 Existence of Injective Extracting Conductors ............. 32  
      3.5.3 Existence of Injective Expanders ......................... 35  
   3.6 Application of Expander Graphs ................................ 37  
   3.7 Notations Used in the Literature ............................... 39  

4 Basic Constructions  
   4.1 Trevisan’s Extracting Conductor ................................ 41  
      4.1.1 Nisan-Wigderson Pseudo-Random Generator .................. 41  
      4.1.2 Making the NW Generator an Injective Extracting Conductor 44  
      4.1.3 Instantiation of Trevisan’s Conductor ..................... 48  
      4.1.4 Expander Graph Construction ............................... 49  
   4.2 Constructing Conductors from Hash Functions ................... 50  
   4.3 Strong Condensing Conductor ................................... 53  
      4.3.1 Reconstructive Extracting Conductors ...................... 53  
      4.3.2 Strong Condensing Conductor Construction .................. 55
1 Introduction

1.1 Unbalanced Bipartite Expander Graphs

There are several functions which have interesting combinatoric properties for applications in cryptographic protocols. *Extractors*, *condensers*, and *conductors* are examples for such functions. In particular, the mentioned functions are all special cases of so-called bipartite expander graphs: A \((K, \gamma)\)-expander graph is a bipartite graph \(G = (V_1, V_2, E)\) such that for every set \(X \subseteq V_1\) with \(|X| \leq K\), the size of its neighbors \(\Gamma(X)\) is at least \(\gamma \cdot |X|\).

In this thesis, we focus on *unbalanced* bipartite expander graphs, where \(|V_1| \gg |V_2|\) and we analyze concrete constructions of such expander graphs. To simplify the analysis, we develop a generalized framework for bipartite expander graphs. It turns out that extractors, condensers, and conductors are special instantiations of our generalized definition. We give a survey of the different expander graph properties, of the achievable parameters, and give composition theorems.

Several constructions of unbalanced bipartite expander graphs are discussed in the literature but unfortunately, the interesting ones have only an asymptotic analysis of the expander graph parameters. Our goal is to give the concrete functions describing the parameter values of the analyzed expander graph constructions and we are mainly interested in the concrete value of the so called *left-degree* of the expander graph, which is the maximal degree of the vertices in \(V_1\).

Unbalanced bipartite expander graphs with small left-degree are interesting for cryptographic applications. One of this applications which uses such expander graphs is the hash function proposed in [MT07]. We give a short overview of this application in Section 1.2.

1.2 Domain Extension for Public Random Functions

In cryptography, functions which take as input a bit string of arbitrary length and return an (almost) random string of fixed size are important for many applications. In general, such a hash function \(\{0, 1\}^* \rightarrow \{0, 1\}^\ell\) is constructed by using a component function \(F : \{0, 1\}^n \rightarrow \{0, 1\}^\ell\) with \(n > \ell\) and embedding this component function into an iterated construction\(^1\) \(H(\cdot)\) resulting in a hash function \(H(R) : \{0, 1\}^* \rightarrow \{0, 1\}^\ell\). In [MT07], they investigate in how to construct a public hash function by using a public random function\(^2\) \(R\) as component function.

The main goal of [MT07] is to construct such a public random function \(R : \{0, 1\}^n \rightarrow \{0, 1\}^\ell\) given public random functions with smaller domain \(\{0, 1\}^m\). They discuss on how to *extend the domain* of the given public random functions from \(\{0, 1\}^m\) to the domain \(\{0, 1\}^n\) (with \(n > m\))

\(^1\)e.g. the CBC or Merkle-Damgård construction

\(^2\)A public random function \(R\) is similar to a secret random function but has a private \((R_{\text{priv}})\) interface where only honest users have access to and a public \((R_{\text{pub}})\) interface where the adversary has access to. Both interfaces \(R_{\text{priv}}\) and \((R_{\text{pub}}\) have the same behavior.
with the help of an efficient construction $C$. But first, we give an overview of the construction $C$, we introduce the notion of *indifferentiable* which is a generalization of *indistinguishable* to systems with public interface and we also define the notion of a *reduction* for public systems.

For a function $F$ or more generally, a system $F$ with a public and a private interface, we write $F = [F_{\text{pub}}, F_{\text{priv}}]$. Further, we denote by $\Delta^D(F, G)$ the *distinguishing advantage* of the distinguisher $D$ in distinguishing the public system $F$ from the (ideal) public system $G$ after doing $k$ queries to the system $F$ and $G$.

Then for a function $\alpha : \mathbb{N} \to \mathbb{R}_{\geq 0}$ and a function $\sigma : \mathbb{N} \to \mathbb{N}$, we say that system $F$ is $(\alpha, \sigma)$-*indifferentiable* from $G$ if there exists a simulator $S$ such that

$$\Delta^D([F_{\text{pub}}, F_{\text{priv}}], [S(G_{\text{pub}}), G_{\text{priv}}]) \leq \alpha(k)$$

for all distinguishers $D$ making at most $k$ queries and $S$ making at most $\sigma(k)$ queries to $G_{\text{pub}}$ when interacting with $D$. Further, we say that a construction $C(\cdot)$ is an $(\alpha, \sigma)$-*reduction*, if

$$\Delta^D([F_{\text{pub}}, C(F_{\text{priv}})], [S(G_{\text{pub}}), G_{\text{priv}}]) \leq \alpha(k).$$

In [MT07], the goal is to construct a reduction $C$ to a public random function $R$ which maps an $n$-bit string to an $\ell$-bit string by using $r + t$ public random functions $F_1, \ldots, F_r : \{0,1\}^m \to \{0,1\}^{\ell \text{pm}}$ and $G_1, \ldots, G_t : \{0,1\}^m \to \{0,1\}^\ell$, where security up to $k = 2^{(1-\epsilon)m} - r$ queries is given for a constant $\epsilon \in (0, 1)$. Further, the construction $C$ uses $r$ efficiently computable functions $E_1, \ldots, E_r : \{0,1\}^n \to \{0,1\}^m$, which we describe later and are for us of special interest.

\[ s \in \{0,1\}^n \]

![Figure 1.1: Construction of public random function $\{0,1\}^n \to \{0,1\}^\ell$](image)

The construction $C$ is illustrated in Figure 1.1 and does the following computations for an input $x \in \{0,1\}^n$:

1. For all $p = 1, \ldots, r$, compute $F_p(E_p(x)) = F_p^{(1)}(E_p(x))|| \cdots ||F_p^{(t)}(E_p(x))$, where $F_p(E_p(x)) \in \{0,1\}^{\ell \text{pm}}$ and $F_p^{(q)}(E_p(x)) \in \{0,1\}^{\ell \text{pm}}$ for all $q = 1, \ldots, t$.  

\[ \text{in Section 2.2.2, the formal definition will be given} \]
2. Compute for all \( q = 1, \ldots, t \) the product \( \mathcal{S}_q := \bigodot_{p=1}^r F_p^{(q)}(E_p(x)) \) where \( \bigodot \) denotes the multiplication in \( GF(2^{om}) \) with \( \rho m \)-bit strings interpreted as elements of the finite field \( GF(2^{om}) \).

3. For all \( q = 1, \ldots, t \), define \( w_q(x) \) being the first \( m \) bits of \( \mathcal{S}_q \).

4. Finally, calculate for each \( q = 1, \ldots, t \) the value \( G_q(w_q(x)) \) and output the sum \( \bigoplus_{q=1}^t G_q(w_q(x)) \).

We explain now on an abstract level why the different stages are needed and what requirements the functions \( E_i \) must fulfill.

We see that if we omit the public random functions \( F_1, \ldots, F_r \), an unbounded adversary could deterministically calculate all outputs \( \bigoplus_{q=1}^t G_q(w_q(x)) \) and possibly find a subset of outputs which are differentiable from a truly random \( l \)-bit string. Hence, the inputs for the public random functions \( G_1, \ldots, G_t \) must contain some randomness.

We discuss now what requirements the function family \( E_1, \ldots, E_r \) must fulfill. First, we require that the adversary cannot find queries \( s \neq s' \in \{0,1\}^n \) such that the construction \( C \) has the same output, i.e. we want to avoid that the adversary can find a collision. To avoid collisions, we require that the function family \( E_1, \ldots, E_r \) is injective, i.e. for all \( s \neq s' \in \{0,1\}^n \), there must exist a \( p \in \{1, \ldots, r\} \) with \( E_p(s) \neq E_p(s') \). Second, the construction must permit simulations of the public random functions \( F_1, \ldots, F_r \) and \( G_1, \ldots, G_t \) given access only to the public interface of a public random function \( R : \{0,1\}^n \rightarrow \{0,1\}^t \). Furthermore, the probability of simulation failure must be small enough to allow security beyond the birthday barrier\(^4\). But if \( E_1, \ldots, E_r \) allow a relatively small number of queries to \( F_1, \ldots, F_r \) for revealing a too large number of values \( w_1(x), \ldots, w_t(x) \), the simulator will possibly fail. To avoid this problem, the function family \( E_1, \ldots, E_r \) must satisfy the following requirement: The family should reveal only a small number of strings in \( \{0,1\}^m \) for a given number of input strings in \( \{0,1\}^n \). Function families being injective and having this restricting property are called input-restricting functions. In [MT07] it is shown why this properties are sufficient to guarantee the security of the construction beyond the birthday barrier. We will give now a precise definition of the input-restricting functions \( E_1, \ldots, E_r \) and in Section 3.6 we will show how to construct such a function family with the help of an unbalanced bipartite expander graphs.

**Definition 1.1** (input-restricting function). [MT07]. Let \( \epsilon = \epsilon(n) \in (0,1) \), \( r = r(n) \), \( \delta = \delta(n) \), \( m = m(n) \) be functions of \( n \) and let \( n > m \), then a family \( \mathcal{F}_n \) of functions \( E_1, \ldots, E_r : \{0,1\}^n \rightarrow \{0,1\}^m \) is called \((n, \delta, \epsilon)\)-input restricting if it satisfies the following two properties:

**Injective:** \( \forall x \neq x' \in \{0,1\}^n \), \( \exists i \in 1, \ldots, r \) such that \( E_i(x) \neq E_i(x') \).

**Input-Restricting:** For all subsets \( \mathcal{M}_1, \ldots, \mathcal{M}_r \subset \{0,1\}^m \) such that \( |\mathcal{M}_1| + \ldots + |\mathcal{M}_r| \leq 2^{(1-\epsilon)m} \), we have

\[
|\{ x \in \{0,1\}^n | E_i(x) \in \mathcal{M}_i \text{ for all } i = 1, \ldots, r \}| \leq \delta \cdot (|\mathcal{M}_1| + \ldots + |\mathcal{M}_r|).
\]

\( \mathcal{F}_n \) is called explicit if \( r(n) \) is polynomial in \( n \) and if \( E_i(\cdot) \) can be computed in \( \text{poly}(n) \) time.

\(^4\)With birthday barrier we mean that the number of queries is up to \( O(2^{n/2}) \).
CHAPTER 1. INTRODUCTION

Choosing \( \mathcal{I}_n = E_1, \ldots, E_r \) being a \((\delta, n, \epsilon)\)-input-restricting function family, the following result was stated in [MT07].

**Theorem 1.2.** Let \( \rho = \lceil n/m + 2 - \epsilon \rceil \) and \( t = \lceil 2/\epsilon - 1 \rceil \). Then the construction \( C \) is an \((\alpha, \sigma)\)-reduction of the public random function \( R : \{0,1\}^n \rightarrow \{0,1\}^t \) to the public random functions \( F_1, \ldots, F_r : \{0,1\}^m \rightarrow \{0,1\}^{t \cdot \rho m} \) and \( G_1, \ldots, G_t : \{0,1\}^m \rightarrow \{0,1\}^t \), where for all \( k \leq 2^{(1-\epsilon)m} - r \),

\[
\alpha(k) \leq 2^t (\delta + 1)^{t + 1} \cdot k^{t + 2} \cdot 2^{-mt} + \frac{1}{2} \cdot t \cdot (\delta + 1) \cdot k \cdot (k + 2r + 1) \cdot 2^{n - \rho m}
\]

and \( \sigma(k) \leq \delta \cdot k \).

In particular, if \( \epsilon \) is constant and \( \delta \) and the cardinality \( r \) are polynomial in \( n \), the above advantage \( \alpha(k) \) is negligible.

Because we want the construction \( C \) be efficient, we require that \( \mathcal{I}_n \) is an explicit input-restricting function family. Especially the requirement of polynomial cardinality \( r \) will give us a strong requirement for the unbalanced expander graphs we are going to construct. Namely, we will need unbalanced bipartite expander graphs which have a small left-degree.

1.3 Contributions

In this thesis, we discuss the following points.

- In Chapter 3, we give a survey over our generalized framework. In particular,
  - in Section 3.1, we introduce a generalized notion of expander graphs and give the formal definition of expander graph properties we use and
  - in Section 3.2, a generalized notion of special combinatorial functions known as extractors, condensers or conductors is given and we call this generalized notion generalized conductors. Furthermore, in Section 3.3, we present strong composition theorems which allow us to build generalized conductors with stronger properties than the underlying conductors.
  - In Section 3.4, we show that our generalized conductors are expander graphs with strong properties.
  - In Section 3.5, we show non-constructively that generalized conductors and expander graphs with strong properties exist, and in particular, with properties interestingly for the application of domain extension of public random functions.

- In Chapter 4, we discuss some concrete basic constructions of generalized conductors which we will use in Chapter 5 to construct stronger generalized conductors with the help of the composition theorems introduced in Section 3.3. In particular in Section 5.4, we present a concrete construction of an expander graph fulfilling in complexity-theoretic terms the requirements needed to get good input-restricting functions and show that this construction is not applicable in practice.

- Finally in Chapter 6, we give a strong impossibility proof which states that it is impossible to construct an expander graph with useful parameters by using a construction based on substring selection.

The used notation and some mathematical preliminaries are introduced in Chapter 2.
2 Mathematical Preliminaries and Notation

2.1 Notation

We use the following notation: With upper-case letters we denote distributions or random variables (RV) and for their concrete values the corresponding lower-case letter. $P_X$ stands for the probabilistic distribution function of the random variable $X$ and $P_X[x]$ is a shorthand for $P[x = X]$. With calligraphic letters ($\mathcal{A}, \mathcal{B}, \ldots$) we denote events or sets. With $[k]$ we denote the set $\{1, 2, \ldots, k\}$.

Let $y$ be an $d$-bit string and $S \subseteq [d]$, then we denote $y|_S$ as the projection or reduction of $y$ to the bits specified by $S$. Further, we denote with $\ln(\cdot)$ the natural logarithm and with $\log(\cdot)$ the logarithm with base $2$.

2.2 Probability and Information Theory

We give now a short overview of the concepts in probability and information theory we use later.

2.2.1 Entropy and Min-Entropy

One of the concepts we need is the entropy function of Shannon which measures the uncertainty of a random variable.

**Definition 2.1 (entropy).** For a discrete random variable $X$ the entropy $H(X)$ is defined as

$$H(X) = \sum_{x \in X} (P_X[x] \cdot \log_2 P_X[x]),$$

where $0 \cdot \log_2 0$ is assumed to be $0$.

For a binary random variable $X$ with bias $p$ we define $h(p) := H(X) = -(p \log p + (1-p) \log(1-p))$, where $h(\cdot)$ is called the binary entropy function. It is easy to see that for the binary entropy the following lemma must hold.

**Lemma 2.2.** For any $\alpha > \beta$ such that $h(\beta) < h(\alpha)$, we have $h(\alpha) < h(\beta) \cdot \frac{\alpha}{\beta}$.

To measure how random a random variable is, we use the min-entropy which is a better measurement for randomness than the entropy function.

**Definition 2.3 (min-entropy).** The min-entropy $H_\infty(X)$ of a distribution $X$ is defined as

$$H_\infty(X) = \min_x \{- \log_2 (P_X[x])\} = -\log_2 (\max_x P_X[x]).$$

For the uniform distribution $U$, we have $H_\infty(U) = H(U)$, but in general, we have $H_\infty(X) \leq H(X)$ for a distribution $X$. 
2.2.2 Distributions

For distributions, we have the notion of support.

**Definition 2.4** (support). The support supp($X$) of a distribution $X$ is the smallest closed set whose complement has probability zero.

**Definition 2.5** (flat distribution). A distribution $X$ is flat if it is uniform over its support $S$, i.e. for every $x \in X$ we have $P_X[x] = \frac{1}{|S|}$.

**Lemma 2.6.** A distribution $X$ has $H_\infty(X) \geq k$ if and only if $X$ is a convex combination of flat distributions on sets of size exactly $2^k$.

**Lemma 2.7** (Chernoff bound). Let $X_1, X_2, \ldots, X_n$ be independent 0/1 random variables and let $\mu$ be the expectation of the sum over this $n$ RVs, then we have

$$P\left[\left|\sum_{i=1}^{n} X_i - \mu\right| > \delta \mu\right] < 2e^{-\frac{n\delta^2}{3}}$$

for $0 < \delta \leq 1$.

**Definition 2.8** (statistical difference). Two distributions $X$ and $Y$ with range $S$ have a statistical difference (or distinguishing advantage) $\epsilon$ if

$$|X - Y| := \max_{D} |P_X[D(X) = 1] - P_Y[D(Y) = 1]| = \max_{A} \left|P_X(x \in A) - P_Y(y \in A)\right|$$

$$= \frac{1}{2} \sum_{s \in S} |P_X(s) - P_Y(s)| = \epsilon,$$

where we maximize over all functions $D: S \rightarrow \{0,1\}$ or all subsets $A \subseteq S$. Function $D$ is often called a distinguisher.

The statistical difference fulfills the triangle inequality.

**Lemma 2.9.** For every distributions $X, Y$ and $Z$ we have

$$|X - Z| \leq |X - Y| + |Y - Z|.$$
2.2. PROBABILITY AND INFORMATION THEORY

**Definition 2.10** (\(\varepsilon\)-close). Distribution \(X\) and \(Y\) with range \(S\) are \(\varepsilon\)-close if the statistical difference between \(X\) and \(Y\) is at most \(\varepsilon\) i.e.

\[
|X - Y| \leq \varepsilon
\]

**Definition 2.11** (\(k\)-source). A distribution \(X\) is a \(k\)-source if \(H_\infty(X) \geq k\). \(X\) is a \((k, \varepsilon)\)-source if it is \(\varepsilon\)-close to some \(k\)-source.

**Lemma 2.12.** Let \(X\) be a distribution on a finite set \(S\). Let \(\text{col}(X)\) the collision probability of \(X\), i.e. the probability that if chosen two elements \(x, y\) independently according to \(X\) we have \(x = y\). If \(\text{col}(X) \leq \frac{1 + 4\varepsilon^2}{|S|}\), then \(X\) is \(\varepsilon\)-close to the uniform distribution on \(|S|\).

**Proof.** To show the lemma, we will use the Cauchy-Schwarz inequality:

\[
\sum_{i=1}^{n} (x_i \cdot y_i) \leq \sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}.
\]

The statistical difference between distribution \(X\) and the uniform distribution \(U\) over \(S\) is defined as

\[
|X - U| = \frac{1}{2} \cdot \sum_{s \in S} \left| \text{P}_X[s] - \frac{1}{|S|} \right|
\]

\[
= \frac{1}{2} \cdot \sum_{s \in S} \left( \left| \text{P}_X[s] - \frac{1}{|S|} \right| \cdot 1 \right)
\]

\[
\leq \frac{1}{2} \cdot \sqrt{\sum_{s \in S} \left( \text{P}_X[s] - \frac{1}{|S|} \right)^2} \cdot \sqrt{\sum_{s \in S} 1^2}
\]

\[
= \frac{\sqrt{|S|}}{2} \cdot \sqrt{\sum_{s \in S} \left( \text{P}_X[s]^2 - \frac{2 \text{P}_X[s]}{|S|} + \frac{1}{|S|^2} \right)}
\]

\[
= \frac{\sqrt{|S|}}{2} \cdot \sqrt{\sum_{s \in S} \text{P}_X[s]^2 - \frac{2 |S|}{|S|^2}} + \frac{|S|}{|S|^2}
\]

\[
= \frac{\sqrt{|S|}}{2} \cdot \sqrt{\sum_{s \in S} \text{P}_X[s]^2 - \frac{1}{|S|}}
\]

where at Step (1) we used the Cauchy-Schwarz inequality and at Step (2) we used the fact that \(\sum_{s \in S} \text{P}_X[s] = 1\).

Note that \(\sum_{s \in S} \text{P}_X[s]^2\) is the collision probability of \(X\) and thus, if we insert \(\sum_{s \in S} \text{P}_X[s]^2 \leq \frac{1 + 4\varepsilon^2}{|S|}\) into the equation, we get

\[
|X - U| \leq \frac{\sqrt{|S|}}{2} \cdot \sqrt{\frac{1 + 4\varepsilon^2}{|S|} - \frac{1}{|S|}}
\]

\[
= \varepsilon.
\]

Hence, \(X\) must be \(\varepsilon\)-close to uniform if \(\text{col}(X) \leq \frac{1 + 4\varepsilon^2}{|S|}\). \(\square\)
CHAPTER 2. PRELIMINARIES AND NOTATION

We state the following fact without giving a proof.

Lemma 2.13. If a random variable $Z$ is not $\epsilon$-close to a distribution with min-entropy $\log(\Lambda/\epsilon)$, then $\exists S: |S| = \Lambda$: $P[Z \in S] > \epsilon$.

If there is not only an algorithm $D$ which can distinguish two distributions but can also non-trivially predict the next output bit given the preceding output bits, we call the algorithm a next-bit predictor. We give now the formal definition of a next-bit predictor.

Definition 2.14 (next-bit predictor). Let $X$ be a distribution over $\{0, 1\}^n$. We call a function $T: \{0, 1\}^n \rightarrow \{0, 1\}$ a next-bit predictor for the distribution $X$ with success $p \geq 1/2$, if

$$P_{i \in [n]} X[T(x_1, x_2, ..., x_{i-1}) = x_i] \geq p,$$

for $x \in X$ with the first $i$ bits set to $x_1, x_2, ..., x_i$.

Every next-bit predictor with success $1/2 + \epsilon$ is also a distinguisher with success $\epsilon$. The converse is also true: Every distinguisher can be transformed into a next-bit predictor but with a loss in the advantage. This fact is stated in the following well-known lemma from Yao.

Lemma 2.15 (Yao). If distribution $X$ over $\{0, 1\}^m$ is not $\epsilon$-close to uniform, then there exists a next-bit predictor $T$ for distribution $X$ with success $1/2 + \epsilon/m$.

Lemma 2.16. Let $Y$ be a distribution over $\{0, 1\}^m$ with min-entropy $H_\infty(Y) \leq \epsilon m$. Then, there exists a next-bit predictor $T : \{0, 1\}^{<m} \rightarrow \{0, 1\}$ for $Y$ with success $1 - \epsilon$.

Proof. Let $Y$ be a distribution $Y = (Y_1, Y_2, ..., Y_m)$ over $\{0, 1\}^m$ with entropy $H(Y) \leq \epsilon m$. Further, let

$$p_{i|y_1, ..., y_{i-1}} := P[Y_i = 1|Y_1 = y_1, ..., Y_{i-1} = y_{i-1}].$$

If $i$ and $y_1, ..., y_{i-1}$ are given, an optimal next-bit predictor $T$ outputs a 1 if $p_{i|y_1, ..., y_{i-1}} > 1/2$ and a 0 otherwise. The next-bit predictor has an error of

$$E_{i \in [m], y \in Y} [\min\{p_{i|y_1, ..., y_{i-1}}, 1 - p_{i|y_1, ..., y_{i-1}}\}].$$

We notice that for a $0 \leq p \leq 1$, we have

$$\min\{p, 1 - p\} \leq \min\{p, 1 - p\} \cdot \log \left( \frac{1}{\min\{p, 1 - p\}} \right) \leq p \log \left( \frac{1}{p} \right) + (1 - p) \log \left( \frac{1}{1 - p} \right) = H(p).$$

Therefore, we have

$$E_{i \in [m], y \in Y} [\min\{p_{i|y_1, ..., y_{i-1}}, 1 - p_{i|y_1, ..., y_{i-1}}\}] \leq E_{i \in [m], y \in Y} [H(p_{i|y_1, ..., y_{i-1}})] = \frac{1}{m} \sum_{i=1}^m H(Y_i|Y_1, Y_2, ..., Y_{i-1}) = \frac{1}{m} H(Y) \leq \epsilon.$$

In particular, we have $H_\infty(X) \leq H(X)$ for all distributions $X$ and therefore $\frac{1}{m} H_\infty(Y) \leq \epsilon$ must hold, too. □
2.3 Estimations for the Binomial Coefficient

Definition 2.17 (binomial coefficient). The binomial coefficient is defined as

\[
\binom{n}{k} := \frac{n!}{(n-k)! \cdot k!}.
\]

According to Stirling, we can approximate the factorial of a number as follows:

Lemma 2.18. For \( n \in \mathbb{N} \), we have

\[
n! > \sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n \cdot e^{1/(12n+1)},
\]

where \( e \) is the Euler constant.

An application of Stirling’s approximation gives as the following upper bound for the binomial coefficient.

Lemma 2.19. For \( n, k \in \mathbb{N} \) with \( k \leq k \), we have

\[
\binom{n}{k} \leq \left(\frac{e \cdot n}{k}\right)^k,
\]

where \( e \) is the Euler constant.

For our calculations we need a special upper bound for the sum over binomial coefficients which we denote in the following lemma.

Lemma 2.20. For \( n \in \mathbb{N} \) and \( 0 \leq \alpha \leq 1/2 \) with \( \alpha n \in \mathbb{N} \) we have

\[
\sum_{i=0}^{\alpha n} \binom{n}{i} \leq 2^{n \cdot h(\alpha)},
\]

with \( h(\alpha) \) being the binary entropy function.

Proof. We have

\[
1 = (\alpha + (1 - \alpha))^n \geq \sum_{i=0}^{\alpha n} \binom{n}{i} \alpha^i (1 - \alpha)^{n-i} \geq (1 - \alpha)^n \sum_{i=0}^{\alpha n} \binom{n}{i} \left(\frac{\alpha}{1 - \alpha}\right)^{\alpha n} = \alpha^{\alpha n} (1 - \alpha)^{(1-\alpha)\alpha n} \sum_{i=0}^{\alpha n} \binom{n}{i},
\]

hence, we have

\[
\sum_{i=0}^{\alpha n} \binom{n}{i} \leq \alpha^{-\alpha n} (1 - \alpha)^{-n(1-\alpha)} = 2^{-n(\alpha \log(\alpha) + (1-\alpha) \log(1-\alpha))} = 2^{n \cdot h(\alpha)}
\]
We will also need a lower bound for the binomial coefficient described in the next lemma.

**Lemma 2.21.** For \( n \in \mathbb{N} \) and \( 0 \leq \alpha \leq 1/2 \) with \( \alpha n \in \mathbb{N} \), we have

\[
\binom{n}{\alpha n} \geq \frac{2^{n-h(\alpha)}}{e \cdot \sqrt{2\pi \alpha(1-\alpha)n}},
\]

with \( h(\alpha) \) being the binary entropy function.

**Proof.** We have

\[
\binom{n}{\alpha n} = \frac{n!}{(\alpha n)!((1-\alpha)n)!},
\]

and applying Stirling’s Approximation \( n! \geq \sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n \cdot e^{1/(12n+1)} \) leads to

\[
\binom{n}{\alpha n} > \frac{\sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n}{\sqrt{2\pi \alpha n} \cdot \sqrt{2\pi (1-\alpha)n} \cdot \left(\frac{\alpha n}{e}\right)^{\alpha n} \cdot \left(\frac{(1-\alpha)n}{e}\right)^{(1-\alpha)n} \cdot \frac{e^{1/(12\alpha+n+1)} e^{1/(12(1-\alpha)n+1)}}{e^{1/(12\alpha+n+1)} e^{1/(12(1-\alpha)n+1)}}}
\]

If we regard at the last factor

\[
\frac{e^{1/(12\alpha+n+1)} e^{1/(12(1-\alpha)n+1)}}{e^{1/(12\alpha+n+1)} e^{1/(12(1-\alpha)n+1)}},
\]

we have that this factor has its minimum for \( \alpha = 0 \) and its minimum is \( 1/e \). Thus,

\[
\binom{n}{\alpha n} > \frac{\sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n}{\sqrt{2\pi \alpha n} \cdot \sqrt{2\pi (1-\alpha)n} \cdot \left(\frac{\alpha n}{e}\right)^{\alpha n} \cdot \left(\frac{(1-\alpha)n}{e}\right)^{(1-\alpha)n} \cdot \frac{e^{1/(12\alpha+n+1)} e^{1/(12(1-\alpha)n+1)}}{e^{1/(12\alpha+n+1)} e^{1/(12(1-\alpha)n+1)}}}
\]

\[
= \frac{1}{e \cdot \sqrt{2\pi \alpha(1-\alpha)n}} \cdot \frac{e^{\alpha n} \cdot e^{-(1-\alpha)n} \cdot (\alpha n)^{\alpha n} \cdot ((1-\alpha)n)^{(1-\alpha)n}}{1}
\]

\[
= \frac{1}{e \cdot \sqrt{2\pi \alpha(1-\alpha)n}} \cdot \frac{n^{\alpha n} \cdot n^{1-\alpha} \cdot (\alpha^{\alpha n} \cdot (1-\alpha)^{(1-\alpha)n})}{1}
\]

\[
= \frac{1}{e \cdot \sqrt{2\pi \alpha(1-\alpha)n}} \cdot \frac{1}{n^{2n(\alpha \log \alpha + (1-\alpha) \log(1-\alpha))}}
\]

\[
= \frac{1}{e \cdot \sqrt{2\pi \alpha(1-\alpha)n}} \cdot 2^{n-h(\alpha)} = \frac{1}{e \cdot \sqrt{2\pi \alpha(1-\alpha)n}} \cdot 2^{n-h(\alpha)},
\]

where at Step (*) we used the definition of the binary entropy function.

\( \square \)

### 2.4 Graph Theory

For a graph \( G \) we use the notation \( G = (V, E) \) where \( V \) is the vertex set and \( E \) is the (multi)set of all edges of graph \( G \). We say \( G \) is a multi graph if there are two or more identical elements in \( E \), i.e. there are pairs of vertices which are connected by more than one edge. In this work, we will only use undirected graphs which means that \( (v_1, v_2) \in E \) implies that we can also go from vertex \( v_2 \) to vertex \( v_1 \) but we do not explicitly insert \( (v_2, v_1) \) into \( E \), too.
undirected (multi) graph $G = (V, E)$ is called bipartite if there exists a partition $V = V_1 \cup V_2$ of the vertex set with $V_1 \cap V_2 = \emptyset$ such that every edge in $E$ is of the form $(v_1, v_2)$ for $v_1 \in V_1$ and $v_2 \in V_2$. We call $G$ balanced if we have $|V_1| = |V_2|$. In the remaining sections we will write $G = (V_1, V_2, E)$ for a bipartite (multi) graph.

An interesting property of a vertex $v$ in $G$ is its degree $d(v)$ which is the number of edges incident to $v$. For bipartite graphs, the concept of the degree can be generalized to a property of the whole graph. Namely, we say that a bipartite graph $G = (V_1, V_2, E)$ has left-degree $D$ if the degree of all $v \in V_1$ is upper bounded by $D$. The right-degree of $G$ is defined analogous.
3 Generalized Unbalanced Bipartite Expander Graphs and Generalized Conductors

In the first two sections of this chapter, we introduce the framework of our generalized unbalanced bipartite expander graphs (short expander graphs) and of our generalized conductors. And in the end of this chapter, we give the relation between our generalized notion and the notion used in the literature. In Section 3.3, we present strong composition theorems which allow us to build generalized conductors with stronger properties than the underlying conductors. Furthermore in Section 3.4, we show that every generalized conductor is a generalized expander graph. In Section 3.5, we prove non-constructively the existence of generalized conductors and expander graphs with strong properties and in Section 3.6 we investigate in the application introduced in Section 1.2 and show how to interpret expander graphs to get an input-restricting function family.

3.1 Generalized Unbalanced Bipartite Expander Graphs

There are several definitions of what a graph has to fulfill to be an expander graph. For example, for all subsets of vertices with a certain size, the number of outgoing edges from the subset has to be bigger than the size of the subset. But we use another definition which is widely accepted. Namely, we will require that for every subset of vertices with size up to a given bound, the number of its neighbors is at least the size of the subset multiplied by an expansion factor. In this work, we require additionally, that an expander graph has to be a bipartite (multi) graph $G = (V_1, V_2, E)$ and we require only good vertex expansion of the left vertex set. Or more formally, for every $X \subset V_1$ (with restricted size), we have $|\Gamma(X)| \geq \gamma \cdot |X|$ where $\Gamma(X)$ are all neighbors of $X$ in $V_2$ and $\gamma$ is called the expansion factor. We give now the formal definition of bipartite expander graphs which has additional restrictions about the size of $\mathcal{X}$ and which is a generalized notion of the commonly used definition because we introduce an additional parameter, namely the lower bound for the set size $|\mathcal{X}|$.

![Figure 3.1: Example for an expander graph: $K_{5,3}$](image)
Definition 3.1 (expander graph). A bipartite (multi) graph $G = (\mathcal{V}_1, \mathcal{V}_2, E)$ with $|\mathcal{V}_1| = N$, $|\mathcal{V}_2| = M$ and left-degree $D$ is an $(N, K_{\text{min}}, K_{\text{max}}) \times (D) \xrightarrow{\gamma} (M)$ expander graph if $|\Gamma(\mathcal{X})| \geq \gamma \cdot |\mathcal{X}|$ for all subsets $\mathcal{X} \subseteq \mathcal{V}_1$ such that $|\mathcal{X}| = K$ and $K \in [K_{\text{min}}, K_{\text{max}}]$, where $\Gamma(\mathcal{X}) \subseteq \mathcal{V}_2$ is the set of neighbors of $\mathcal{X}$.

It is easy to see that $\gamma \leq D$. One example of such an expander graph is the $K_{5,3}$ graph. Every subset of the left vertex set has three neighbors and we get a minimal vertex expansion of $3/5$. However, we would like such a graph to be sparse and having small left-degree. Furthermore, we will only consider unbalanced expander graphs with $|\mathcal{V}_1| \gg |\mathcal{V}_2|$ which is a reason why we are only interested in the expansion property of the left vertex set. In particular, whenever we will talk about expander graphs or expanders we mean actually unbalanced bipartite expander graphs. We introduce now the notion of injectivity for an expander graph which will be later in Section 3.6 of special interest for the application of expander graphs.

Definition 3.2 (injective expander graph). Let $\Gamma(v, i)$ be the $i$th neighbor of vertex $v$. An $(N, K_{\text{min}}, K_{\text{max}}) \times (D) \xrightarrow{\gamma} (M)$ expander graph $G = (\mathcal{V}_1, \mathcal{V}_2, E)$ is injective if

$$\forall v \neq v' \in \mathcal{V}_1 : \exists i \text{ such that } \Gamma(v, i) \neq \Gamma(v', i).$$

Every expander graph can be interpreted as a function. Let $G = (\mathcal{V}_1, \mathcal{V}_2, E)$ with $|\mathcal{V}_1| = 2^n$, $|\mathcal{V}_2| = 2^m$ and $D = 2^d$ be an expander graph, then we can define a function $F_G : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$, where $F_G(x, i)$ is the $i$-th neighbor of $x$. Formally:

Lemma 3.3. An $(2^n, K_{\text{min}}, K_{\text{max}}) \times (2^d) \xrightarrow{\gamma} (2^m)$ expander graph $G = (\mathcal{V}_1, \mathcal{V}_2, E)$ is a function $F_G : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$, for which every distribution $X$ over $\{0,1\}^n$ with $K_{\text{min}} \leq |\text{supp}(X)| \leq K_{\text{max}}$, we have $|\text{supp}(F_G(X, U_d))| \geq \gamma \cdot |\text{supp}(X)|$.

Proof. Because not every left vertex has $2^d = D$ neighbors, we double some edges until every left vertex has degree $D$. Though we get a multi graph, it is still a valid expander graph. Furthermore, we know that for every set $\mathcal{X} \subseteq \mathcal{V}_1$ with $K_{\text{min}} \leq |\mathcal{X}| \leq K_{\text{max}}$ we have $|\Gamma(\mathcal{X})| \geq \gamma \cdot |\mathcal{X}|$. Let $X$ be a distribution with $K_{\text{min}} \leq |\text{supp}(X)| \leq K_{\text{max}}$. We set $\mathcal{X} := \text{supp}(X)$ and because $K_{\text{min}} \leq |\text{supp}(X)| \leq K_{\text{max}}$, we have that $|\Gamma(\text{supp}(X))| \geq \gamma \cdot |\text{supp}(X)|$. Together with the definition of function $F_G$, this leads to $|\text{supp}(F_G(X, U_d))| \geq \gamma \cdot |\text{supp}(X)|$. 

We are interested in expander graphs having big vertex sets $\mathcal{V}_1$ and $\mathcal{V}_2$ and thus, it is not practicable to describe the expander graph with a $|\mathcal{V}_1| \times |\mathcal{V}_2|$-matrix. Hence, we want that the function $F_G : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ runs in $\text{poly}(n)$ time for all inputs and we can therefore, efficiently build the expander graph. We call expander graphs explicit if they are such functions $F$ computable in $\text{poly}(n)$ time.

Definition 3.4 (explicit expander graph). An $(N, K_{\text{min}}, K_{\text{max}}) \times (D) \xrightarrow{\gamma} (M)$ expander graph $G = (\mathcal{V}_1, \mathcal{V}_2, E)$ is explicit if function $F_G : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ can be evaluated in $\text{poly}(n)$ time for every input.

Often, it is easier to see an expander graph as a function as described in Lemma 3.3 and to do the analysis for this function rather than for the graph. In particular, we will consider functions which have stronger properties besides being an expander graph and afterwards, we

---

$^1$ $K_{5,3}$ denotes the complete bipartite graph $G = (\mathcal{V}_1, \mathcal{V}_2, E)$ with $|\mathcal{V}_1| = 5$ and $|\mathcal{V}_2| = 3$
show how to construct further expander graphs which rely on these additional properties of the underlying functions.

Therefore, we will introduce in the next section a family of special combinatorial functions, called \textit{generalized conductors}, and show in Section 3.4 that every generalized conductor is an expander graph.

3.2 Generalized Conductors

Building an expander graph is a non-trivial problem, and often achieved by first constructing special combinatorial functions which extract some random bits from a \(k\)-source and afterwards, transforming this functions into an expander graph. This functions are known in the literature as \textit{condensers, conductors or extractors}. Each of this combinatorial functions have their own properties and theorems but actually, we point out that they have a lot in common and thus, we develop a generalized framework and call the generalized combinatorial function a \textit{generalized conductor}. Condensers, conductors and extractors are a special instantiation of our conductors and in Section 3.7 we will show how to instantiate our generalized conductors to get condensers, conductors and extractors. We call our generalized conductors just \textit{conductors} and point out which version we mean when confusion to the original conductors is possible.

We introduce now our generalized conductor framework and start by giving a definition of the most general form of a conductor.

\textbf{Definition 3.5} (generalized conductor). A function \(C : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m\) is an \((n,k_{\text{min}},k_{\text{max}}) \times (d) \rightarrow_\epsilon (m,k_m(k'))\) \textit{conductor} if for every distribution \(X\) over \(\{0,1\}^n\) with min-entropy \(k' \in [k_{\text{min}},k_{\text{max}}]\), the output distribution \(C(X,U_d)\) is \(\epsilon\)-close to a distribution with \(k_m(k')\) min-entropy, where \(k_m\) is a monotone growing function in \(k'\).

The second input in \(\{0,1\}^d\) is often called the \textit{seed}. Note that the distribution \(X\) over \(\{0,1\}^n\) can have more than \(k_{\text{max}}\) min-entropy but the function \(k_m\) is only defined for \(k' \in [k_{\text{min}},k_{\text{max}}]\). Hence, we get only the guarantee that up to \(k_m(k_{\text{max}})\) min-entropy will be extracted even so the input min-entropy is larger than \(k_{\text{max}}\).

Sometimes a stronger property is needed, namely that even if one reveals the \(d\) random input bits, the output bits contain still the condensed randomness:

\textbf{Definition 3.6} (strong conductor). An \((n,k_{\text{min}},k_{\text{max}}) \times (d) \rightarrow_\epsilon (m,k_m(k'))\) conductor \(C\) is called \textit{strong} if \(C(X,U_d) \circ U_d\) is \(\epsilon\)-close to a distribution with \(k_m(k') + d\) min-entropy.

As for expander graphs, we introduce a notion of injectivity for our conductors.

\textbf{Definition 3.7} (injective conductor). We call an \((n,k_{\text{min}},k_{\text{max}}) \times (d) \rightarrow (m,k_m(k))\) conductor \textit{injective} if

\[ \forall x \neq x' \in \{0,1\}^n, \exists y \in \{0,1\}^d \text{ such that } C(x,y) \neq C(x',y). \]

As for expander graphs, we define \textit{explicit} conductors.

\textbf{Definition 3.8} (explicit conductor). Let \(k_{\text{min}},k_{\text{max}},d,m\) and \(k_m(k')\) be functions of \(n\). Then, \(C\) is an \textit{explicit} \((n,k_{\text{min}},k_{\text{max}}) \times (d) \rightarrow (m,k_m(k'))\)-conductor if \(C(\cdot,\cdot)\) can be computed in \(\text{poly}(n)\) time\(^2\).

\(^2\)More formally, \(C\) is a family of conductors \(C_n\) which are efficient in respect to the security parameter \(n\), where all parameters depend on \(n\).
Ideally, one can hope that a conductor can extract all \( k' \) random bits from the first input and that also all \( d \) random bits are transformed to the output. But not every conductor is able to achieve this. The gap between actual obtained min-entropy in the output and ideally possible min-entropy is called the entropy loss.

**Definition 3.9 (entropy loss).** The entropy loss \( \Delta \) of an \( (n, k_{\text{min}}, k_{\text{max}}) \times (d) \rightarrow (m, k_m(k')) \) conductor \( C \) is a function \( \Delta(k') := k' + d - k_m(k') \) which is defined for \( k' \in [k_{\text{min}}, k_{\text{max}}] \). For \( C \) being strong, the entropy loss is \( \Delta(k') := k' - k_m(k') \).

If the conductor preserves at least the min-entropy of the input distribution, we say that it is a **condensing** conductor\(^3\).

**Definition 3.10 (condensing conductor).** An \( (n, k_{\text{min}}, k_{\text{max}}) \times (d) \rightarrow (m, k_m(k')) \) conductor \( C \) is condensing if \( k_m(k') \geq k' \) for \( k' \in [k_{\text{min}}, k_{\text{max}}] \).

A stronger requirement is to have a conductor which retains all the input min-entropy and not only the min-entropy of the first input. We call such conductors **lossless**.

**Definition 3.11 (lossless conductor).** An \( (n, k_{\text{min}}, k_{\text{max}}) \times (d) \rightarrow (m, k' \mapsto k' + d) \) conductor \( C \) is lossless.

It is easy to see that the following is true.

**Lemma 3.12.** Every strong condensing conductor is a lossless conductor.

An other special family of conductors is where the \( m \)-bit output is actually an \( (m, \epsilon) \)-source, i.e. almost truly random, if the input has min-entropy \( k_{\text{max}} \). This conductors are known as **extracting** conductors.

**Definition 3.13 (extracting conductor).** An \( (n, k_{\text{min}}, k_{\text{max}}) \times (d) \rightarrow (m, k_m(k')) \) conductor \( \text{Ext} \) is called an extracting conductor if \( k_m(k_{\text{max}}) = m \).

An important fact is that a non-trivial extracting conductor cannot be lossless where with non-trivial we mean that \( k_{\text{max}} \geq 1 \). There is always an entropy loss and its lower bound is shown in [RT00]:

\(^3\)We use the term **condensing** as a synonym for **preserving**. In the literature, there are functions called **condensers** and our conductors are a generalization of those condensers. More details about condensers can be found in [TUZ01].
Lemma 3.14 (lower bound of min-entropy). Every non-trivial extracting \((n, k_{\min}, k_{\max}) \times (d) \rightarrow_{\varepsilon} (m, k_m(k'))\) conductor has an entropy loss \(\Delta(k') = d + k' - k_m(k') \geq 2 \log(1/\varepsilon) - O(1)\) for \(k' \in [k_{\min}, k_{\max}]\).

Additionally, [RT00] states the the lower bound for the seed length:

Lemma 3.15 (lower bound of seed length). Every non-trivial extracting \((n, k_{\min}, k_{\max}) \times (d) \rightarrow_{\varepsilon} (m, k_m(k'))\) conductor has seed length \(d \geq \log(n - k_{\max}) + 2 \log(1/\varepsilon) - O(1)\) for \(k' \in [k_{\min}, k_{\max}]\).

In Section 3.5.2 we will show non-constructively the existence of extracting conductors with short seed length and which reach the lower bound of the entropy loss up to a constant term.

We introduce now another special form of a conductor \(C\). Let \(C\) be a function \(C : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^{m-t}\) which outputs \(t\) blocks of length \(m\) and for all but a fraction of \(\sigma\) inputs there is at least one \(m\)-bit block in the output which has the needed conductor properties. We call this functions \(\sigma\)-somewhere conductors because the output contains somewhere a block with the wished properties. We give now the formal definition.

Definition 3.16 (somewhere conductor). A function \(C : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^{m-t}\) is a \(\sigma\)-somewhere \((n, k_{\min}, k_{\max}) \times (d) \rightarrow_{\varepsilon} (m, k_m(k'))\) conductor if for all \(k'\)-source \(X\) over \(\{0, 1\}^n\), there exists a selection function \(I_{P_X} : \{0, 1\}^n \rightarrow \{1, \ldots, t\} \cup \{\bot\}\) such that \(P_{I_{P_X}(X)[\bot]} \leq \sigma\) and \(P_{C(i)(X, U_d)\mid I_{P_X}(X)=i} \) is a \((k_m(k'), \varepsilon)\)-source for all \(i = 1, \ldots, t\) with \(P_{I_{P_X}(X)(i)} > 0\), where \(C(X, U_d) = C(1)(X, U_d)|\ldots||C(t)(X, U_d)\) and \(C(i)(X, U_d) \in \{0, 1\}^m\) for all \(i = 1, \ldots, t\).

I.e., for every but a fraction of \(\sigma\) inputs \(x \in \{0, 1\}^n\) of a \(k'\)-source \(X\), the selection function \(I_{P_X}(X)\) associates an index \(i\) to it such that \(C(i)(x, U_d)\) is \(\varepsilon\)-close to a \(k_m(k')\)-source. Further, it is easy to see that every conductor is also a 0-somewhere conductor. A special case of a somewhere conductor is when we use extracting conductors as basic building blocks \(C(i)\). Let \(Ext(k)\) be an extracting \((n, k, k) \times (d(k)) \rightarrow_{\varepsilon} (m(k), m(k))\) conductor for min-entropy \(k\), which needs \(d(k)\) random bits as seed and outputs an almost random \(m(k)\)-bit string, where \(m(\cdot)\) is a monotone growing function. Depending on \(k\), the length of the seed and the output of \(Ext(k)\) changes. We will use \(k\) different extracting conductors \(Ext(1), Ext(2), \ldots, Ext(k)\) for every min-entropy in \(\{0, 1, \ldots, k\}\). Let \(d\) be the longest seed length needed and ignore the last few bits if \(Ext(i)\) does not need it. We denote with \(m\) the maximal output length \(m(k_{\max})\) and will extend every output to the length \(m\) by padding zeros, i.e. we get an \((n, i, i) \times (d) \rightarrow_{\varepsilon} (m, m(i))\) conductor \(C(i)(x, y) = Ext(i)(x, y)|00\ldots0\). For every distribution \(X\) we set the selection function \(I_{P_X}(\cdot) = i\), where \(2^i \leq H_\infty(X) \leq 2^{i+1}\) must hold. Note that \(P_{I_{P_X}(X)}[\bot] = 0\) and hence, \(\sigma = 0\) because for every \(i\)-source \(X\) we can associate a conductor \(C(i)\) which returns an \(m(i)\)-source for all inputs \(x \in X\).

Lemma 3.17. If we use extracting conductors as \(C(i)\)'s as described above, we get a 0-somewhere \((n, 0, k) \times (d) \rightarrow_{\varepsilon} (m \cdot k, m(k'))\) conductor.

We point out that we do not analyze expander graph constructions which use this special kind of somewhere conductors. Instead, we will present an alternative construction in Section 5.4 which leads to expander graphs with smaller left-degree.
3.3 Composition Theorems for Conductors

In this section, we present different constructions to combine conductors to get a new conductor with stronger properties. Some compositions were already introduced in the literature for special combinatorial functions like extractors and condensers \[RRV99,\ TUZ01\]. We present improved versions of this different compositions by also considering injectivity and set them in a more generalized framework to fit into our generalized conductor notion.

3.3.1 Conductor Cascading

We start by defining a cascading operator \(\circ\) for conductors as follows:

**Definition 3.18 (cascading).** Let \(C_1 : \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{m_1}\) and \(C_2 : \{0,1\}^n \times \{0,1\}^{d_2} \to \{0,1\}^{m_2}\) be two functions, then we define the cascading \((C_2 \circ C_1) : \{0,1\}^n \times \{0,1\}^{d_1+d_2} \to \{0,1\}^{m_2}\) by

\[
(C_2 \circ C_1)(x; y_1, y_2) = C_2(C_1(x, y_1), y_2).
\]

An illustration of cascading two functions \(C_1\) and \(C_2\) is given in Figure 3.3. If we cascade two conductors with special properties, than their cascading \((C_2 \circ C_1)\) is also a conductor which inherits the properties of the basic \(C_1\) and \(C_2\) conductors. We state this fact in the next lemma.

**Lemma 3.19 (conductor cascading).** Given two conductors \(C_1\) and \(C_2\) where

- \(C_1\) is a (strong) (injective) \((n, k_{\text{min}}, k_{\text{max}}) \times (d_1) \to \epsilon_1 (m_1, k_1(k'))\) conductor and
- \(C_2\) is a (strong) (injective) \((m_1, k_1(k_{\text{min}}), k_1(k_{\text{max}})) \times (d_2) \to \epsilon_2 (m_2, k_2(k'))\) conductor,

their cascading \((C_2 \circ C_1)\) is a (strong) (injective) \((n, k_{\text{min}}, k_{\text{max}}) \times (d_1 + d_2) \to \epsilon_1 + \epsilon_2 (m_2, k_2(k'))\) conductor.

**Proof.** First, we show the non-strong case: Let \(X \subseteq \{0,1\}^n\) be a \(k'\)-source with \(k' \in [k_{\text{min}}, k_{\text{max}}]\). We know that \(C_1(X, U_{d_1})\) is \(\epsilon_1\)-close to a distribution \(Y\) with \(k_1(k')\) min-entropy. Further, \(C_2(Y, U_{d_2})\) is \(\epsilon_2\)-close to a distribution having \(k_2(k'(k'))\) min-entropy. Therefore, by applying the triangle inequality for statistical difference (Lemma 2.9), we have that \(C_2(C_1(X, U_{d_1}), U_{d_2})\) is at least \((\epsilon_1 + \epsilon_2)\)-close to a distribution with \(k_2(k_1(k'))\) min-entropy.
In this section, we present a concatenation operator $||$ which is based on the concatenation introduced in [RRV99]. The proof is adapted for our generalized conductors.

**Definition 3.20** (conductor concatenation). Let $C_1 : \{0,1\}^n \times \{0,1\}^{d_1} \rightarrow \{0,1\}^{m_1}$ and $C_2 : \{0,1\}^n \times \{0,1\}^{d_2} \rightarrow \{0,1\}^{m_2}$ be two functions. Then we define their *concatenation* $(C_1||C_2) : \{0,1\}^n \times \{0,1\}^{d_1+d_2} \rightarrow \{0,1\}^{m_1+m_2}$ as

$$C(x, (y_1, y_2)) = C_1(x, y_1)||C_2(x, y_2).$$

In Figure 3.4, we illustrate the concatenation of the functions $C_1$ and $C_2$.

![Conductor Concatenation Diagram](image)

Figure 3.4: Conductor concatenation

If we require that $C_1$ and $C_2$ are strong conductors with according parameters, than we can construct a conductor with much less entropy loss than the original conductors have. We state this property in the next lemma and we will make use of it in the Sections 5.1 and 5.2.

**Lemma 3.21** (conductor concatenation). Let $s > 0$. Given two conductors $C_1$ and $C_2$, where

- $C_1$ is a strong extracting $(n, k_{\min}, k_{\max}) \times (d_1) \rightarrow_{\epsilon_1} (m_1, k_1(k'))$ conductor with entropy loss $\Delta_1(k') = k' - k_1(k')$ for $k' \in [k_{\min}, k_{\max}]$ and $k_1(\cdot)$ a monotone growing function with $k_1(k_{\max}) = m_1$,
and $C_2$ is a strong extracting $(n, \Delta_1(k_{\text{min}}) - s, \Delta_1(k_{\text{max}}) - s) \times (d_2) \rightarrow (m_2, k_2(k'))$ conductor with entropy loss $\Delta_2(k') = k'' - k_2(k')$ for $k'' \in [\Delta_1(k_{\text{min}}) - s, \Delta_1(k_{\text{max}}) - s]$ and $k_2(\cdot)$ a monotone growing function $k_2(\cdot)$ with $k_2(\Delta_1(k_{\text{max}}) - s) = m_2$.

Then their concatenation $(C_1 || C_2)$ is a strong $(n, k_{\text{min}}, k_{\text{max}}) \times (d_1 + d_2) \rightarrow (m_1 + m_2, k_m(k'))$ conductor with error $\epsilon = \left(\frac{1}{\log_2 s}\right) \cdot \epsilon_1 + \epsilon_2$, entropy loss $\Delta(k') = \Delta_2(\Delta_1(k') - s) + s$ and $k_m(k') = k_1(k') + k_2(k')$. Furthermore, the final conductor $(C_1 || C_2)$ is injective if at least one of the two conductors $C_1$ and $C_2$ is injective.

**Proof.** First, we show why $\Delta(k') = \Delta_2(\Delta_1(k') - s) + s$. When we apply the first extracting conductor $C_1$ we get an entropy loss of at most $\Delta_1(k')$. We show that for all $s > 0$, there are still $\Delta_1(k') - s$ min-entropy in $X$ which has not been extracted. The second conductor $C_2$ extracts $\Delta_1(k') - s - \Delta_2(\Delta_1(k') - s)$ min-entropy from $X$. Therefore, the remaining entropy loss is just $\Delta_2(\Delta_1(k') - s) + s$.

It remains to show that a $k'$-source $X$ has still $\Delta_1(k') - s$ “unused” min-entropy in it after applying $C_1$. We define a set of bad inputs for which the output’s probability of $C_1$ is smaller than $2k_1(k')$ by a factor of $2^s$. Let $X$ be a $k'$-source and let $BAD$ be the set of pairs $(u, z) \in \{0, 1\}^{d_1} \times \{0, 1\}^{m_1}$ such that

$$P[C_1(X, u) = z] < 2^{-(k_1(k') + s)}. \quad (3.3.1)$$

We will show now that for all good pairs, i.e. every $(u, z) \notin BAD$, the distribution $P_{X[C_1(X, u) = z]}$ has still at least $\Delta_1(k') - s$ min-entropy and thus, $C_2$ can be applied without any problems.

Let $(u, z) \notin BAD$. Then for every $x \in X$ with $C_1(x, u) = z$ we have

$$P_{X[C_1(X, u) = z]}[x] = \frac{P_X[x]}{P[C_1(X, u) = z]} \leq \frac{2^{-k'}}{2^{-(k_1(k') - s)}} = 2^{-(\Delta_1(k') - s)}$$

where at the last step we used $\Delta_1(k') = k' - k_1(k')$ because $C_1$ is a strong conductor. Hence, the distribution $P_{X[C_1(X, u) = z]}$ has min-entropy of at least $\Delta_1(k') - s$.

Furthermore, for every good pair, we have that the conditional distribution of $(U_{d_2}, C_2(X, U_{d_2}))$ given that $(U_{d_1}, C_1(X, U_{d_1})) = (u, z)$ is $\epsilon_2$-close to $U_{d_2} \circ Y_2$, where $Y_2 \subseteq \{0, 1\}^{m_2}$ is some distribution with min-entropy $k_1(k')$.

We show now that the probability of $(u, z)$ being a pair in $BAD$ is relatively small. For all $(u, z) \in BAD$ we have that $P[(U_{d_1}, C_1(x, U_{d_1})) = (u, z)] < 2^{-d_1 - (k_1(k') + s)}$. Let $Y_1 \subseteq \{0, 1\}^{m_1}$ be a distribution with min-entropy $k_1(k')$. Then, we have for every $(u, z) \in BAD$

$$P[\text{Y}_1 = (u, z)] = 2^{-d_1 - k_1(k')} \geq 2^s \cdot P[(U_{d_1}, C_1(x, U_{d_1})) = (u, z)] \Rightarrow P[(U_{d_1}, Y_1) \in BAD] \geq 2^s \cdot P[(U_{d_1}, C_1(x, U_{d_1})) \in BAD].$$

By using the definition of the statistical difference we get for the distance between $U_{d_1} \circ Y_1$ and $U_{d_1} \circ C_1(X, U_{d_1})$

$$|U_{d_1} \circ Y_1 - U_{d_1} \circ C_1(X, U_{d_1})| \geq P[(U_{d_1}, A) \in BAD] - P[(U_{d_1}, C_1(x, U_{d_1})) \in BAD] \geq (2^s - 1) \cdot P[(U_{d_1}, C_1(x, U_{d_1})) \in BAD] \quad (3.3.2)$$
Additionally, because $C_1$ is a strong conductor with error at most $\epsilon_1$, we have

$$\epsilon_1 \geq |U_{d_1} \circ Y_1 - U_{d_1} \circ C_1(X, U_{d_1})|$$

and hence by combining (3.3.2) and (3.3.3),

$$P[(U_{d_1}, C_1(x, U_{d_1})) \in BAD] \leq \frac{\epsilon_1}{2^n - 1}.$$

Finally, it remains to show that $C_1(X, U_{d_1}) \circ U_{d_1} \circ C_2(X, U_{d_2}) \circ U_{d_2}$ is $(\frac{\epsilon_1}{1 - 2^{-\frac{3}{2}}} + \epsilon_2)$-close to $Y_1 \circ U_{d_1} \circ Y_2 \circ U_{d_2}$.

We know that $C_1(X, U_{d_1}) \circ U_{d_1}$ is $\epsilon_1$-close to $Y_1 \circ U_{d_1}$ because $C_1$ is strong. Furthermore, we showed above that the probability of bad inputs for $C_2$ is just $\frac{\epsilon_1}{2^n - 1}$ and for the other $1 - \frac{\epsilon_1}{2^n - 1}$ fraction of inputs we know that $C_2(X, U_{d_2}) \circ U_{d_2}$ is $\epsilon_2$-close to $Y_2 \circ U_{d_2}$ because $C_2$ is a strong conductor. Overall, $C_1(X, U_{d_1}) \circ U_{d_1} \circ C_2(X, U_{d_2}) \circ U_{d_2}$ is $(\epsilon_1 + \frac{\epsilon_1}{1 - 2^{-\frac{3}{2}}} + \epsilon_2)$-close to $Y_1 \circ U_{d_1} \circ Y_2 \circ U_{d_2}$ because of the triangle inequality for the statistical difference and because $(1 - \frac{\epsilon_1}{2^n - 1}) \cdot \epsilon_2 \leq \epsilon_2$. Reforming leads to $\epsilon_1 + \frac{\epsilon_1}{1 - 2^{-\frac{3}{2}}} + \epsilon_2 = \frac{\epsilon_1}{1 - 2^{-\frac{3}{2}}} + \epsilon_2$.

Furthermore, if at least one of the two conductors is injective, the overall output must be injective because already a subpart of it is injective. 

We can also define a different version of the conductor concatenation according to [RRV99] which states a concatenation for “non-strong” extracting conductors. We will not use this kind of concatenation in our constructions discussed in the later sections. But for completeness, we state also this concatenation which is interesting if one wants to concatenate non-strong conductors and we use the operator $\parallel$ to clarify the distinction to the operator of Definition 3.20.

**Definition 3.22** (adapted conductor concatenation). Let $C_1 : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^{m_1}$ and $C_2 : \{0,1\}^{n+d_1} \times \{0,1\}^{d_2} \rightarrow \{0,1\}^{m_2}$ be two functions. Then we denote their **concatenation** ($C_1 \parallel C_2$) : $\{0,1\}^n \times \{0,1\}^{d_1+d_2} \rightarrow \{0,1\}^{m_1+m_2}$ as

$$(C_1 \parallel C_2)(x, (y_1, y_2)) = C_1(x, y_1)||C_2((x, y_1), y_2).$$

Note that the difference is by applying $C_2$ to both inputs $(x, y_1)$ of $C_1$ rather than just to $x$.

Also for this kind of operation, we can state properties about the concatenation if the basic conductors have the according parameters.

**Lemma 3.23** (adapted conductor concatenation). Let $s > 0$. Given two conductors $C_1$ and $C_2$, where

- $C_1$ is an extracting $(n, k_{\min}, k_{\max}) \times (d_1) \rightarrow_{\epsilon_1} (m_1, k_1(k'))$ conductor with entropy loss $\Delta_1(k') = k' + d_1 - k_1(k')$ for $k' \in [k_{\min}, k_{\max}]$ and $k_1(\cdot)$ a monotone growing function with $k_1(k_{max}) = m_1$,

- and $C_2$ is an extracting $(n + d_1, \Delta_1(k_{\min}) - s, \Delta_1(k_{\max}) - s) \times (d_2) \rightarrow_{\epsilon_1} (m_2, k_2(k''))$ conductor with entropy loss $\Delta_2(k'')$ for $k'' \in [\Delta_1(k_{\min}) - s, \Delta_1(k_{\max}) - s]$, and $k_2(\cdot)$ a monotone growing function with $k_2(\Delta_1(k_{\max}) - s) = m_2$. 

21
Then their concatenation \((C_1 \| C_2)\) is an \((n, k_{\min}, k_{\max}) \times (d_1 + d_2) \rightarrow (m_1 + m_2, k_m(k'))\) conductor with error \(\epsilon = (\frac{1}{2^k_2}) \cdot \epsilon_1 + \epsilon_2\) and entropy loss \(\Delta(k') = \Delta_2(\Delta_1(k') - s) + s\). Furthermore, the final conductor \((C_1 \| C_2)\) is injective if at least one of the two conductors \(C_1\) and \(C_2\) is injective.

Proof sketch. The proof is analogue to the proof of Lemma 3.21. The differences are that the set \(BAD\) contains now elements \(z \in \{0, 1\}^{m_1}\) such that
\[
P[C_1(X, U_{d_1}) = z] < 2^{-(k_1(k') + s)}
\]
and that we get
\[
P_{X \circ U_{d_1}|C_1(X, U_{d_1}) = z}[x] = \frac{P_X(x) \cdot 2^{-d_1}}{P[C_1(X, U_{d_1}) = z]} \leq 2^{-(\Delta_1(k') - s)}.
\]
The remaining parts of the proof is done as in the proof of Lemma 3.21 but without concatenating the used truly random strings, respectively concatenating the uniform distributions.

### 3.3.3 Constructing Somewhere-Conductors by Conductor Cascading

We show how now to get a somewhere conductor \(C\) by cascading a strong extracting conductor \(C_1 : \{0, 1\}^n \times \{0, 1\}^{d_1} \rightarrow \{0, 1\}^{d_2}\) with a strong conductor \(C_2 : \{0, 1\}^n \times \{0, 1\}^{d_2} \rightarrow \{0, 1\}^m\). This cascading differs from the cascading defined in Section 3.3.1 and will be of special interest in Section 5.4 where we give a concrete construction of an expander graph with small left-degree. The construction of somewhere-conductors introduced in this section and the proof in Section 3.3.3 are based on [NT99, BJST03, MT07].

First, we introduce a new notation; For a string \(x \in \{0, 1\}^n\), let \(x_{[a,b]}\) be the string consisting of the bits \(x_a, x_{a+1}, \ldots, x_{b-1}, x_b\) with extra 0’s appended to make \(|x_{[a,b]}| = n\). If \(b < a\) then we set \(x_{[a,b]} = 0^n\). We define \(C : \{0, 1\}^n \times \{0, 1\}^{d_1} \rightarrow \{0, 1\}^{n \cdot (d_1 + d_2 + m)}\) such that \(C(x, y) = C^{(1)}(x, y)||\ldots||C^{(n)}(x, y)\), where for all \(1 \leq i \leq n\), we have
\[
z_1^{(i)} := y \quad z_2^{(i)} := C_1(x_{[i,n]}, z_1^{(i)}) \quad z_3^{(i)} := C_2(x_{[i,i-1]}, z_2^{(i)})
\]
and set \(C^{(i)}(x, y) := z_1^{(i)}||z_2^{(i)}||z_3^{(i)} \in \{0, 1\}^{d_1 + d_2 + m}\).

In Figure 3.5, an illustration of the construction of the \(i\)th output \(C^{(i)}(x, y)\) is given, where the gray areas mark the padding with zeros.

**Lemma 3.24.** Let \(\nu > 0\) be given, and \(C\) being constructed as above. If \(C_1\) is a strong \((n, 0, d_2 - a_1) \times (d_1) \rightarrow \epsilon_1 (d_2, k' \rightarrow k' + a_1)\) conductor, and \(C_2\) is a strong \((n, 0, k_{\max}) \times (d_2) \rightarrow \epsilon_2 (m, k' \rightarrow k' + a_2)\) conductor, then \(C\) is a \(\sigma\)-somewhere \((n, 0, d_2 - a_1 + k_{\max} + s) \times (d_1) \rightarrow \epsilon_1 + \epsilon_2 (n \cdot (m + d_1 + d_2), k' \rightarrow k' + a)\) conductor with \(\sigma = 7n \cdot 2^{-\nu/3}\) and \(a = \min \{a_1, a_1 + a_2\} + d_1 - \nu\).

**Proof of Lemma 3.24.** Let \(X\) be a \(k\)-source with \(k \leq d_2 - a_1 + k_{\max} + \nu\). We distinguish two cases.

**Case 1.** \(H_\infty(X) = k = \hat{k} + \nu\) with \(\hat{k} \leq d_2 - a_1\).

The distribution \(X\) has less than \(d_2 - a_1\) min-entropy up to a small summand \(\nu\). Therefore, we
3.3. COMPOSITION THEOREMS FOR CONDUCTORS

Figure 3.5: Construction of $C^{(i)}(x, y)$

just set $P_{I_{PX}}(X|i = 1) = 1$ and get $[Z_1^{(1)}, Z_2^{(1)}] = [U_{d_1}, C^{(1)}(X, U_{d_1})]$ which is a $(d_1+k+a_1, \epsilon_1)$-source. Because $\nu > 0$, we have that $[Z_1^{(1)}, Z_2^{(1)}]$ is a $(d_1+k+a_1-\nu, \epsilon_1)$-source, too. When we append $Z_3^{(1)}$, we get the distribution $[Z_1^{(1)}, Z_2^{(1)}, Z_3^{(1)}]$ which has at least the min-entropy of $[Z_1^{(1)}, Z_2^{(1)}]$ and thus is also a $(k+a_1+d_1-\nu, \epsilon_1)$-source.

Case 2. $H_\infty(X) = k = d_2 - a_1 + \hat{k} + \nu$ with $\hat{k} \leq k_{max}$.

For this case, we will show that there exists a selector function $I_{PX} : \{0, 1\}^n \rightarrow \{1, ..., n\} \cup \{\bot\}$ such that

1. $P_{I_{PX}}(X|\bot) = \sigma \leq 7n \cdot 2^{-\nu/3}$
2. If $P_{I_{PX}}(X|X_{[1,i-1]}[i, x_{[1,i-1]}]) > 0$, then $H_\infty(X_{[i,n]}|I_{PX}(X) = i \wedge X_{[1,i-1]} = x_{[1,i-1]}) \geq d_2 - a_1$
3. $H_\infty(X_{[1,i-1]}|I(X) = i) \geq \hat{k}$

Proof for Point 1. Let $\zeta_1 = 2\zeta_2$, $\zeta_2 = 2\zeta_3$ and $\zeta_3 = 2^{-\nu/3}$. Further, we will define the selection function $I(X)$ similar to [NT99]. Let the function $f : \{0, 1\}^n \rightarrow \{1, ..., n\}$ return the last $i$ on input $x \in X$ such that the remaining block $x_{[i,n]}$ is still random enough, i.e.

$$P[X_{[i,n]} = x_{[i,n]}|X_{[1,i-1]} = x_{[1,i-1]}] \leq (\zeta_2 - \zeta_3) \cdot 2^{-(d_2 - a_1)}. \quad (3.3.4)$$

Some splitting points $i$ are rare and might lead to a strange behavior. Let $BAD$ be the set of all bad $x$. We define $x$ to be in $BAD$ if $f(x) = i$ and

- $P_{x' \in X}[f(x') = i] \leq \zeta_1$
- $P_{x' \in X}[f(x') = i|x_{[1,i-1]} = x'_{[1,i-1]}] \leq \zeta_2$
- $P_{x' \in X}[x'_i = x_i|x_{[1,i-1]} = x'_{[1,i-1]}] \leq \zeta_3$

We define $I_{PX}(X)$ such that it filters this bad cases:

$$I_{PX}(x) = \begin{cases} \bot & x \text{ is bad} \\ f(x) & \text{otherwise}. \end{cases}$$
Putting everything together, we get \( P(x \in BAD) = P_{IP_X}(X)[\perp] = n(\zeta_1 + \zeta_2 + \zeta_3) \leq 7n \cdot 2^{-\nu/3}. \)

**Proof for Point 2.** To show Point 2, we use the following lemma which was shown in Appendix C of [NT99].

**Lemma 3.25.** For any \( i \) and \( x_{[1,i-1]} \), if

\[
P_{x' \in X}[IP_X(x') = i | x'_{[1,i-1]} = x_{[1,i-1]]} > 0
\]

then \( P_{x' \in X}[IP_X(x') = i | x'_{[1,i-1]} = x_{[1,i-1]}] > \zeta_2 - \zeta_3 \), where \( \zeta_i \) are defined as above.

For any \( x \) such that \( IP_X(x) = i \), we have

\[
P[X_{[1,n]} = x_{[1,n]} | IP_X(X) = i \wedge X_{[1,i-1]} = x_{[1,i-1]}] \leq \frac{P[X_{[1,n]} = x_{[1,n]} | X_{[1,i-1]} = x_{[1,i-1]}]}{2 - (d_2 - a_1)} \]

\[
\leq \frac{P[IP_X(x) = i | X_{[1,i-1]} = x_{[1,i-1]}]}{(\zeta_2 - \zeta_3) \cdot 2^{-(d_2 - a_1)}} = 2^{-(d_2 - a_1)}
\]

In the first step we used \( P[X|Y] \leq P[X]/P[Y] \), at Step (3.3.5) we applied the requirement (3.3.4) of function \( f \) and in (3.3.6) we used Lemma 3.25.

Hence, it immediately follows that \( H_\infty(X_{[1,n]} = x_{[1,n]} | IP_X(X) = i \wedge X_{[1,i-1]} = x_{[1,i-1]}]) \geq d_2 - a_1. \)

**Proof for Point 3.** We fix an \( x \) with \( IP_X(x) = i \) and get

\[
P[X_{[1,i-1]} = x_{[1,i-1]}] = \frac{P[X_{[1,n]} = x_{[1,n]}]}{\sum_{k=0}^{\infty} 2^{-(d_2 - a_1) + k + \nu}} \leq \frac{P[X_i = x_i | X_{[1,i-1]} = x_{[1,i-1]}] \cdot P[X_{i+1,n} = x_{[i+1,n]} | X_{[1,i-1]} = x_{[1,i-1]}]}{2^{-(d_2 - a_1) + k + \nu}} \leq \frac{2^{-(d_2 - a_1) + k + \nu}}{\zeta_3 \cdot (\zeta_2 - \zeta_3) 2^{-(d_2 - a_1)}} \leq \frac{2^{-k - \nu}}{\zeta_3 \cdot (\zeta_2 - \zeta_3)}
\]

where in Step (3.3.7) we used the fact that \( H_\infty(X) = d_2 - a_1 + k + \nu \) and in Step (3.3.8) we applied the Requirement (3.3.4) of \( f(x) = i \) and that \( x \notin B \). For the following steps, we need a lemma from [NT99]:

**Lemma 3.26.** For any \( i \), if \( P_{x \in X}[IP_X(x) = i] > 0 \), then \( P_{x \in X}[IP_X(x) = i] \geq \zeta_1 - \zeta_2 - \zeta_3 \), where \( \zeta_i \) are defined as above.
Finally, to prove $H_\infty(X_{[1,i-1]}|IP_X(X) = i) \geq \hat{k}$, we show that $P[X_{[1,i-1]} = x_{[1,i-1]}|IP_X(x) = i]$ is at most $\hat{k}$:

$$P[X_{[1,i-1]} = x_{[1,i-1]}|IP_X(x) = i] \leq \frac{P[X_{[1,i-1]} = x_{[1,i-1]}]}{P[IP_X(x) = i]} \leq \frac{2^{-k-\nu}}{\zeta \cdot (\zeta_2 - \zeta_3) \cdot (\zeta_1 - \zeta_2 - \zeta_3)} \cdot 2^{-k-\nu} = 2^{-\nu/3} \cdot (2 \cdot 2 - 2^{1/3} - 2^{1/3}) \cdot (4 \cdot 2 - 2^{1/3} - 2^{1/3}) = 2^{-k}. \tag{3.3.10}$$

At (3.3.10) we used $P[X|Y] \leq P[X]/P[Y]$ and for Step (3.3.11) we applied Lemma 3.26 and Equation (3.3.9).

According to Point 2, we can conclude that for all $x_{[1,n]}$ with the property that

$$P_{IP_X(X)|X_{[1,i-1]}=i,x_{[1,i-1]}} > 0$$

we have that the distribution $P_{Z_1^{(i)}Z_2^{(i)}IP_X(X)=i,X_{[1,i-1]}=x_{[1,i-1]}}$ is $\epsilon$-close to a $(d_1 + d_2)$-source because the strong conductor $C_1$ has a $(d_2 - a_1)$-source as input. Further, we know that $P_{Z_1^{(i)}Z_2^{(i)}IP_X(X)=i}$ is a $(d_1 + d_2 + \hat{k}, \epsilon_1)$-source because of Point 3. Applying the conductor $C_2$ to the $\hat{k}$-source $X_{[1,i-1]}$ leads to being $P_{Z_1^{(i)}Z_2^{(i)}C_2(X_{[1,i-1]}),Z_2^{(i)}IP_X(X)=i}$ a $(d_1 + d_2 + \hat{k} + a_2, \epsilon_1 + \epsilon_2)$-source. Using the precondition $k = d_2 - a_1 + \hat{k} + \nu$ we get that $P_{Z_1^{(i)}Z_2^{(i)}Z_3^{(i)}IP_X(X)=i}$ is a $(k + a_1 + a_2 + d_1 - \nu, \epsilon_1 + \epsilon_2)$-source.

Putting the two analyzed cases together leads to a $\sigma$-somewhere conductive $(n, 0, k) \times (d_1) \rightarrow (m, k' \leftarrow k' + a)$ conductor with

- $k = d_2 - a_1 + k_2 + \nu$
- $a = \min\{a_1 + d_1 - \nu, a_1 + a_2 + d_1 - \nu\}$
- $\epsilon = \max\{\epsilon_1, \epsilon_1 + \epsilon_2\} = \epsilon_1 + \epsilon_2$
- $\sigma = \max\{7n \cdot 2^{-\nu/3} = 7n \cdot 2^{-\nu/3}\}$

which concludes the proof.

### 3.4 Constructing Expander Graphs From Conductors

As already mentioned in the previous sections, conductors are expander graphs. In this section, we show how to achieve this transformation from a conductor to an expander graph. We know that every $(2^n, K_{min}, K_{max}) \times (2^d) \rightarrow (2^n, \gamma)$ expander graph $G = (V_1, V_2, E)$ can be interpreted as a function $F_G : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ which calculates the neighbors of a vertex $v \in V_1$. If we compare function $F_G$ with an $(n, k_{min}, k_{max}) \times (d) \rightarrow (m, k_m(k'))$ conductor $C$, we see that the conductor has the same form $\{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ and $x \in \{0, 1\}^n$ interpreted as a vertex $v \in V_1$, the conductor $C$ would calculate the $i$th neighbor
of \( v \) where \( i \) is randomly chosen from \( \{0, 1\}^d \). As we will show later in this section, the lower and upper bound \( K_{\min} \) and \( K_{\max} \) and the min-entropy bounds \( k_{\min} \) and \( k_{\max} \) are highly correlated, namely we will have \( K_{\min} = 2^{k_{\min}} \) and \( K_{\max} = 2^{k_{\max}} \). Note that in general, the converse is not true. Not every expander graph is a conductor.

In the following, we will show how to transform a conductor into an expander graph and also the special case where the conductor is a \text{somewhere} conductor.

### 3.4.1 Transforming Conductors into an Expander Graph

In this section, we show how to generally transform a (strong) conductor into an expander graph. We start with the case where we have a strong conductor \( C \): To get an expander graph, we interpret every input \( x \in \{0, 1\}^n \) as a vertex \( v_1 \in V_1 \), and every input \( y \in \{0, 1\}^d \) as the label for the \( y \)-th neighbor of \( v_1 \). The output \( C(x, y) \circ U_d \in \{0, 1\}^{m+d} \) of the strong conductor \( C \) will then describe a vertex \( v_2 \in V_2 \) being the \( y \)-th neighbor of vertex \( v_1 \).

The exact relation between the parameters of the strong conductor \( C \) and the achieved expander graph \( G \) is given in the next Theorem 3.27.

**Theorem 3.27.** Let \( C : \{0, 1\}^n \times \{0, 1\}^d \to \{0, 1\}^m \) be a function. The bipartite graph \( G = (V_1, V_2, E) \) with \( |V_1| = 2^n \), \( |V_2| = 2^{m+d} \) and

\[
(v_1, (y, v_2)) \in E \iff C(v_1, y) = v_2
\]

is an (explicit) (injective) \((2^n, K_{\min}, K_{\max}) \times (D) \) \( \to (2^{m+d}) \) expander graph with \( K_{\min} = 2^{k_{\min}} \), \( K_{\max} = 2^{k_{\max}} \), left-degree \( D = 2^d \) and expansion factor \( \gamma = (1-\epsilon)2^{d+\alpha} \) if \( C \) is a strong (explicit) (injective) \((n, k_{\min}, k_{\max}) \times (d) \) \( \to (m, k' \to k' + \alpha) \) conductor. Note that \( \alpha \) can be negative.

**Proof.** Let \( X \) be a \( k' \)-source with \( k' \in [k_{\min}, k_{\max}] \) and \( |X| \in [2^{k_{\min}}, 2^{k_{\max}}] \). According to the strong conductor property, we know that the distribution \( A = C(X, U_d) \circ U_d \) is \( \epsilon \)-close to a distribution \( A' \) on \( \{0, 1\}^m \times \{0, 1\}^d \) with \( H_\infty(A') \geq k' + \alpha + d \). The set of neighbors is \( \Gamma(X) = supp(A) \). This leads to

\[
\epsilon \geq |A - A'| \geq \sum_{w \in \Gamma(X)} |A(w) - A'(w)| = 1 - \sum_{w \in \Gamma(X)} A'(w) \geq 1 - |\Gamma(X)| \cdot 2^{-(k' + \alpha + d)}
\]

By rearranging terms we get the wished expansion property: \( |\Gamma(X)| \geq (1-\epsilon)2^{k' + \alpha + d} \geq (1-\epsilon)2^d \cdot |X| \). Furthermore, it is clear from the transformation that if the conductor \( C \) is explicit or injective than so the expander graph.

Even if the conductor is not strong, we get an expander graph but with a worse expansion factor.

**Corollary 3.28.** Let \( C : \{0, 1\}^n \times \{0, 1\}^d \to \{0, 1\}^m \) be a function. The bipartite graph \( G = (V_1, V_2, E) \) with \( |V_1| = 2^n \), \( |V_2| = 2^{m+d} \) and

\[
(v_1, (y, v_2)) \in E \iff C(v_1, y) = v_2
\]

is an (explicit) (injective) \((2^n, K_{\min} = 2^{k_{\min}}, K_{\max} = 2^{k_{\max}}) \times 2^d \to (2^{m+d}, \gamma = (1-\epsilon)2^n) \) expander graph if \( C \) is an (explicit) (injective) \((n, k_{\min}, k_{\max}) \times (d) \to (m, k' \to k' + \alpha) \) conductor. Note that \( \alpha \) can be negative.

**Proof sketch.** The proof is analogue to the proof of Theorem 3.27. Just use the distribution \( A = C(X, U_d) \) instead, i.e. do not append \( U_d \) to the output.  

\[\square\]
3.5. PROBABILISTIC EXISTENCE PROOFS

3.4.2 Transforming Somewhere Conductors into an Expander Graph

In this section, we show how to transform a \( \sigma \)-somewhere conductor into an expander graph. This special case of a conductor is of special interest because as we will see later in Section 5.4, if we first apply the somewhere conductor composition of Section 3.3.3 to special conductors and second, transform the received somewhere conductor to an expander graph, we get a much smaller left-degree than if we had directly transformed the special conductors to an expander graph.

**Lemma 3.29.** If \( C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{tm} \) is a \( \sigma \)-somewhere \((n,k_{\min},k_{\max}) \times (d) \to \epsilon \) \((tm,k' \mapsto k' + a)\) conductor with \( \sigma < 1 \), then graph \( G = (V_1, V_2, E) \) with \( V_1 = \{0,1\}^n, V_2 = \{0,1\}^m \) and

\[
(x, z) \in E \quad \text{if} \quad \exists i \in [t], y \in \{0,1\}^d : C^{(i)}(x, y) = z
\]

is a \((2^n, 2^{k_{\min}}, 2^{k_{\max}}) \times D \to (2^{tm}, \gamma = 2^a(1 - \epsilon))\)-expander graph with left-degree \( D = t \cdot 2^d \).

**Proof.** The proof is similar to the proof of Theorem 3.27. Let \( X \) be a \( k' \leq k_{\max} \) and \( |X| \leq 2^{k_{\max}} \). Let \( I_{PX} : \{0,1\}^n \to \{1,\ldots,t\} \cup \{\perp\} \) be the selection function according to the distribution \( X^4 \). Let \( i \) be such that \( P_{I_{PX}}(X)(i) > 0 \) and let \( A \) be a \((k' + a)\)-source which is \( \epsilon \)-close to \( P_{C^{(i)}(X,U_d)}|I_{PX}(X) = i \). We define the set of neighbors as \( \Gamma(X) = supp(A) \). This leads to

\[
\epsilon \geq |Pr_{C^{(i)}(X,U_d)}|I_{PX}(X) = i - A(w)| \geq \sum_{w \in \Gamma(X)} |Pr_{C^{(i)}(X,U_d)}|I_{PX}(X) = i(w) - A(w)|
\]

\[
= 1 - \sum_{w \in \Gamma(X)} A(w) \geq 1 - |\Gamma(X)| \cdot 2^{-(k' + a)}
\]

By rearranging terms we get the wished expansion property: \( |\Gamma(X)| \geq (1 - \epsilon)2^{k' + a} \geq (1 - \epsilon)2^a|X| \).

3.5 Probabilistic Existence Proofs of Conductors and Expanders

In this section, we prove non-constructively the existence of injective (lossless) conductors and injective extracting conductors with short seed length and almost ideal entropy loss. The existence of such conductors is often stated in the literature but hardly ever proven. Especially the result we will give in Section 3.5.1 seems to never been shown in a publication. In Section 3.5.3, we show non-constructively the existence of an injective expander graph for an arbitrary expansion factor. All the existence proofs in this section are done with the help of the so-called Probabilistic Method, where the existence is implied if the probability for the non-existence is strictly smaller than 1. But all this proofs are non-constructively and it is still an open question if one can give an explicit construction of such good conductors and expanders.

**General Structure of the Proofs.** We show the existence of such almost ideal injective conductors and injective expanders by a non-constructive probabilistic existence proof. The structure of the different proofs is quite similar: We assume that a randomly chosen function with the same domain and range as a conductor has not the wished conductor properties.

\footnote{A possible construction of \( I_{PX}(X) \) is given in the proof of Lemma 3.24}
CHAPTER 3. EXPANDERS AND CONDUCTORS

Then, we show that the probability for not being a conductor with this properties is strictly smaller than 1, which implies the existence of the wished conductor. In particular, we will show the existence of the wished conductors for the special case, where the input distribution $X$ is a flat $k$-source. This restriction to flat sources is sufficient because every $k$-source is a convex combination of flat $k$-sources, and hence, the proof stated for flat $k$-sources implies the validity for general $k$-sources. For the expander graph existence proof, we assume that a randomly chosen graph has not the wished expansion properties. Then, we show that the probability of being a badly chosen graph is strictly smaller than 1. Hence, there is a non-zero probability that a graph with the wished expansion properties must exist.

More detailed, we will show in Section 3.5.1 that for a randomly chosen function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, we have

$$P[C \text{ not injective}] + P[C \text{ not a conductor}] < 1$$

and in Section 3.5.2 we show accordingly,

$$P[C \text{ not injective}] + P[C \text{ not an extracting conductor}] < 1$$

and finally, in Section 3.5.3 we show for a randomly chosen graph $G$ that

$$P[G \text{ not injective}] + P[G \text{ not an expander graph}] < 1.$$ 

In the first two cases, we will show that $P[\text{not injective}] < 1/2$ and the corresponding second probability is strict smaller than $1/2$, and for the third case we will show that $P[\text{not injective}] < 2/3$ and the second summand is strict smaller than $1/3$. Hence, the overall probability must be strict smaller than 1.

**Injectivity.** In particular, the probability for not being injective is always calculated in the same way. We give now the general approach and instantiate it in the according sections.

First, recall that injectivity means that there exist no two distinct values $x, x' \in \{0,1\}^n$ for which

$$C(x,0) | C(x,1) | \cdots | C(x,2^d-1) = C(x',0) | C(x',1) | \cdots | C(x',2^d-1)$$  \hspace{1cm} (3.5.1)

is true, where $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$. Let us assume that $C$ is not injective. Then, there must be a collision $C(x,u) = C(x',u)$ for a $u \in \{0,1\}^d$. We interpret the left-hand side and the right-hand side of Equation (3.5.1) as a string of length $m \cdot 2^d$. In general, the probability for having at least one collision between two strings is $q^{2^d} - r$, where $q$ is the number of possible strings and $r$ the length of a string. Here, the length of a string is $r = m2^d$ and there are at most $q = 2^n$ different strings because we have $2^n$ different values in $\{0,1\}^n$. This gives for the collision probability

$$P[C \text{ not injective}] = P[\text{a collision occurs}] \leq 2^{2n} \cdot 2^{-m2^d}. \hspace{1cm} (3.5.2)$$

We will show in the according Sections 3.5.1, 3.5.2 and 3.5.3 that this probability of not being injective is smaller than $1/2$ respectively smaller than $2/3$.

---

5 For the expander graph, interpret graph $G$ as a function as described in Lemma 3.3.
3.5. PROBABILISTIC EXISTENCE PROOFS

Conductor Properties. For the proofs in Section 3.5.2 and in Section 3.5.2 we take the following approach to bound the probability of not being a (lossless) conductor respectively an extracting conductor. We will assume the existence of a testing set \( T \subseteq \{0,1\}^m \) which allows us to distinguish the output of \( C(X,U_d) \) from a \( k_m(k') \)-source with success \( > \epsilon \), with \( X \) being a flat \( k' \)-source.

Then, we fix a flat \( k' \)-source and the testing set \( T \) and define a new random variable such that we get an upper bound for the probability with the help of Chernoff’s bound (Lemma 2.7). Afterwards, we sum up over all possible distributions \( X \) having min-entropy in the interval \([k_{min},k_{max}]\) and over all possible testing sets \( T \) to get the final union bound for the probability.

We start by the non-constructively existence proof of an injective (lossless) conductor.

3.5.1 Existence of Injective (Lossless) Conductor

In this section, we prove non-constructively the existence of the following conductor.

**Theorem 3.30.** For every \( n, k_{min} \leq k_{max} \leq n \) and \( \epsilon \in (0,1) \), there exists an injective \((n,k_{min},k_{max}) \times (d) \rightarrow (m,k_{m}(k'))\) conductor \( C \) with \( k_{m}(k') = k' + d - q \) for a constant \( q \geq 0 \) and \( k' \in [k_{min},k_{max}] \), seed length \( d = \log(n) + \log(1/\epsilon) + O(1) \) and output length \( m = k_m(k_{max}) + \log(1/\epsilon) + o(1) \).

Note that \( C \) can be lossless (setting \( q = 0 \)), i.e. the output min-entropy is \( k_m(k') = k' + d \) with output length \( m = k_{max} + d + \log(1/\epsilon) + O(1) \).

**Proof.** To show the existence of such an injective conductor by showing that

\[
P[C \text{ not a conductor}] + P[C \text{ not injective}] < 1,
\]

we will fix the min-entropy to a value \( k' \) and show afterwards, that for every min-entropy \( k' \in [k_{min},k_{max}] \) we get a probability smaller than 1. We set \( k_m := k_m(k') = k' + d - q \) for a \( q \geq 0 \) and assume that \( m' := k_m(k') + \log 1/\epsilon + a \) and \( d = \log(n) + \log(1/\epsilon) + b \) for constants \( a \) and \( b \). Note that here, \( m' \) depends on \( k' \) and describes the first \( m' \) bits of the output. We will show that already this first \( m' \) bits of the output of length \( m \) has \( k_m(k') \) min-entropy and we can finally set the output length to \( m \) for all input min-entropies \([k_{min},k_{max}]\) because enlarging the output does not reduce the output min-entropy. Furthermore, we define \( A = 2^a \), \( B = 2^b \), \( N = 2^n \), \( K' = 2^k \) and \( M' = 2^m \). Hence, \( M' = 2^{k_m+\log 1/\epsilon + a} = \frac{K_m A}{\epsilon} = \frac{K'DA}{Q \epsilon} \) for \( K_m = 2^k \), \( D = 2^d \) and \( Q = 2^q \).

We will assume that there exists no such conductor \( C \) fulfilling Theorem 3.30, or in other words,

\[
\exists \ k'-source \ X : \forall \ k_m-source \ Y \subseteq \{0,1\}^m : |C(X,U_d) - Y| > \epsilon,
\]

which is equivalent to the following requirement by applying Lemma 2.13:

If \( C(X,U_d) = Z \) is not \( \epsilon \)-close to a distribution with min-entropy \( k_m = \log(|T|/\epsilon) \), then \( \exists \ T \subseteq \{0,1\}^m \) with \( |T| = \epsilon K_m = \epsilon DK'/Q \), s.t. \( P[C(X,U_d) \in T] > \epsilon \) and \( X \) being a flat \( k' \)-source.

Therefore, we have

\[
P[C(X,U_d) \in T] = \frac{|\{(x,u) \in X \times U_d | C(x,u) \in T\}|}{K'D} > \epsilon.
\]
Let event $B$ be such that
\[ |\{(x, u) \in X \times U_d \mid C(x, u) \in T\}| > \epsilon K'D.\]

We define now a new random variable $X_{x,u}$: It is 1 if $C(x, u) \in T$ and 0 otherwise. We know that $P[X_{x,u} = 1] = \frac{|T|}{M} = \frac{eK_m}{AK_m/\epsilon} = \frac{\epsilon^2}{A}$. Hence, the expected value of the sum $\sum_{x,u} X_{x,u}$ is $\mu = K'D \cdot \frac{|T|}{M} = \frac{K'D\epsilon^2}{A}$.

This gives for the probability of occurring event $B$

\[
P[B] = P\left[\sum X_{x,u} > \epsilon K'D\right] \leq P\left[\left|\sum X_{x,u} - \mu\right| > \epsilon K'D - \mu\right] = P\left[\left|\sum X_{x,u} - \mu\right| > \left(\frac{\epsilon K'D}{\mu} - 1\right) \cdot \mu\right]
\]

and we set
\[
\delta := \frac{\epsilon K'D}{\mu} - 1 = \frac{\epsilon K'DA}{K'D\epsilon^2} - 1 = \frac{A}{\epsilon} - 1.
\]

Using the Chernoff bound of Lemma 2.7

\[
P\left[\left|\sum X_{x,u} - \mu\right| > \delta \mu\right] < 2e^{-\frac{\mu \delta^2}{3}} = 2^{-\gamma(\mu \delta^2)} \quad \text{for a constant } \gamma
\]

gives the upper bound

\[
2^{-\gamma(\mu \delta^2)} = 2^{-\gamma(DK'\frac{2}{\pi} (\frac{A}{\epsilon} - 1)^2)} = 2^{-\gamma DK'(A - \epsilon)^2} \leq 2^{-\gamma DK'},
\]

where at the last step, we fixed $A = 2$ such that $A - \epsilon$ lies in the interval $(1, 2)$ and the term maximizes for $A - \epsilon = 1$ and thus, we can replace $(A - \epsilon)^2$ with 1.

We regard now all possible flat sources $X$ having min-entropy in $[k_{\text{min}}, k_{\text{max}}]$ and hence, $|X| \in [K_{\text{min}}, K_{\text{max}}]$, where $K_{\text{min}} = 2^{k_{\text{min}}}$ and $K_{\text{max}} = 2^{k_{\text{max}}}$. We calculate the union bound of the probability by summing up over all possible distributions $X$ and all possible sets $T$. We get
\[ \Pr[C \text{ not a conductor}] \leq \sum_{i=K_{\text{min}}}^{K_{\text{max}}} \binom{N}{i} \left( \frac{M'}{|T|} \right) \cdot 2^{-\gamma D_i} \]

\[ \leq \sum_{i=K_{\text{min}}}^{K_{\text{max}}} \binom{N}{i} \left( \frac{M'}{|T|} \right) \cdot 2^{-\gamma D_i} \]

\[ \leq \sum_{i=K_{\text{min}}}^{K_{\text{max}}} \binom{N}{i} \left( \frac{M'}{|e_i D/Q|} \right) \cdot 2^{-\gamma D_i} \]

\[ \leq \sum_{i=K_{\text{min}}}^{K_{\text{max}}} \binom{N}{i} \left( \frac{e A}{\epsilon^2} \right) \cdot 2^{-\gamma D_i} \]

\[ \leq \sum_{i=K_{\text{min}}}^{K_{\text{max}}} \binom{N}{i} \cdot \left( \frac{e A}{\epsilon^2} \right) \cdot 2^{-\gamma D_i} \]

\[ = \sum_{i=K_{\text{min}}}^{K_{\text{max}}} \left( 2^n \cdot 2^{e D (\log e + \log A + 2 \log(1/e))} \cdot 2^{-\gamma D} \right)^i < 1/2 \]

where at Step (1) we used \( M' = K' DA \) and \( \left( \frac{e A}{\epsilon^2} \right)^{e D/Q} \leq \left( \frac{e A}{\epsilon^2} \right)^{e D} \). Knowing that \( \log A = 1 \) and inserting the value for \( d \) leads to \( D = 2^{\log n + \log(1/e) + b} = \frac{n B}{\epsilon} \) and gives the requirement:

\[ \sum_{i=1}^{K_{\text{max}}} \binom{N}{i} \left( 2^n \cdot 2^{e D (\log e + \log A + 2 \log(1/e))} \cdot 2^{-\gamma D} \right)^i < 1/2 \]

To fulfill this requirement, it is sufficient to show that the next requirement holds:

\[ 2^n (1 + B (\log e + 1 + 2 \log(1/e) - \frac{\gamma B}{\epsilon})) < \frac{1}{4} \]

(1)

because

\[ \sum_{i=1}^{K_{\text{max}}} \binom{1}{i} < \sum_{i=1}^{\infty} \binom{1}{i} = \frac{1}{2} \]

where we used \( \sum_{i=1}^{\infty} \frac{1}{p^i} = \frac{1}{1-p} - 1 \) for a \( 0 < p < 1 \).

Requirement I is equivalent to the requirement

\[ n \left( 1 + B \left( \log e + 1 + 2 \log(1/e) - \frac{\gamma B}{\epsilon} \right) \right) < -2. \]
We choose $B > \frac{e^{(\log e + 1 + 2\log e + 1)}}{\gamma} > 3$ which leads to $\omega < -2$ and thus

$$\omega n < -2n \leq -2,$$

which is fulfilled if $n > 0$. Thus, Requirement I is satisfied.

We show now that $P[C \text{ not injective}]$ is smaller than $1/2$. Inserting $2^d = \frac{nB}{\epsilon}$ into Equation (3.5.2) gives us the requirement

$$P[C \text{ not injective}] \leq 2^{2n - \frac{m'nB}{\epsilon}} < \frac{1}{2}. \quad \text{(II)}$$

Requirement II is equivalent to $2n - \frac{m'nB}{\epsilon} + 1 < 0$. We know that $\epsilon \in (0, 1)$ and if we choose $B > 2$ then Requirement II holds for all $n, m' > 0$.

Putting everything together, we have

$$P[C \text{ not an injective conductor}] \leq \sum_{i=1}^{\infty} \left( \frac{1}{4} \right)^i + \frac{1}{2} = \frac{1}{3} + \frac{1}{2} < 1.$$ 

We showed that for an output length $m'$ depending on $k'$ where $k'$ is the input min-entropy, we get an injective conductor with $k_m = k_m(k')$ output min-entropy. We can set now the output length for all $k' \in [k_{min}, k_{max}]$ to $m := m'(k_{max})$ because $m'(k') \leq m'(k_{max})$ and when the conductor output contains $k_m(k')$ min-entropy in the first $m'(k')$ output bits, it will also contain $k_m(k')$ min-entropy in the (possibly) longer output length $m$.

Therefore, there must exist with non-zero probability an injective $(n, k_{min}, k_{max}) \times (d) \rightarrow_\epsilon (m, k_m(k'))$ conductor $C$ with $k' \in [k_{min}, k_{max}]$, seed length $d = \log n + \log(1/\epsilon) + O(1)$ and output length $m = k_m(k_{max}) + \log(1/\epsilon) + O(1).$

$\square$

### 3.5.2 Existence of Injective Extracting Conductors

A special case which can not be shown by the proof in Section 3.5.1 is where the conductor is extracting. Therefore, we present a separate proof for this special case in this section. We will show non-constructively the existence of the following conductor:

**Theorem 3.31.** For every $n, k_{min} \leq k_{max} \leq n$ and $\epsilon \in (0, 1)$, there exists an injective extracting $(n, k_{min}, k_{max}) \times (d) \rightarrow_\epsilon (m, k_m(k'))$ conductor $C$ with $k' \in [k_{min}, k_{max}]$, seed length $d = \log(n - k_{min}) + 2\log(1/\epsilon) + O(1)$ and output length $m = k_{max} + d - 2\log(1/\epsilon) + O(1)$.

Note that this extracting conductor has optimal entropy loss up to a constant term according to Lemma 3.14.

**Proof.** As in Section 3.5.1, we fix the input min-entropy to a value $k'$ and generalize it afterwards for the case where the min-entropy lies in the interval $[k_{min}, k_{max}]$. For the output length, we assume that $m' = k_m(k')$. Note that here, $m'$ depends on $k'$ and describes the first $m'$ output bits which are almost randomly distributed. We denote $N = 2^n$, $M' = 2^{m'}$, $D = 2^d$ and $K' = 2^{k'}$. Let $X$ be a flat $k'$-source with support $S$, where $|S| = K'$. We assume that $C$ is not an extracting conductor and therefore there must exist a testing set $T$ such that

$$|P[C(X, U_d) \in T] - P[U_{m'} \in T]| > \epsilon.$$ 

32
3.5. PROBABILISTIC EXISTENCE PROOFS

We know that \( P[U_{m'} \in \mathcal{T}] = \frac{|T|}{M'} \) and

\[
P[C(X, U_d) \in \mathcal{T}] = \frac{|\{(x, u) \in S \times \{0, 1\}^d \mid C(x, u) \in \mathcal{T}\}|}{K'D}
\]

hence, we have

\[
\left| \frac{|\{(x, u) \in S \times \{0, 1\}^d \mid C(x, u) \in \mathcal{T}\}|}{K'D} - P[U_{m'} \in \mathcal{T}] \right| > \epsilon.
\]

Let event \( \mathcal{B} \) be such that

\[
\left| \{(x, u) \in S \times \{0, 1\}^d \mid C(x, u) \in \mathcal{T}\} - K'D \cdot \frac{|T|}{M'} \right| > \epsilon \cdot K'D
\]

We define now a new random variable \( X_{x,u} \): It is 1 if \( C(x, u) \in \mathcal{T} \) and 0 otherwise. We know that \( P[X_{x,u} = 1] = \frac{|T|}{M'} \). Thus, the expected value of the sum \( \sum_{x,u} X_{x,u} \) is \( \mu = K'D \cdot \frac{|T|}{M'} \).

Hence, we get

\[
P[\mathcal{B}] = P \left[ \left| \left\{(x, u) \mid C(x, u) \in \mathcal{T} \right\} - \mu \right| > \epsilon K'D \right]
\]

\[
= P \left[ \left| \sum_{x,u} X_{x,u} - \mu \right| > \epsilon K'D \right]
\]

\[
= P \left[ \left| \sum_{x,u} X_{x,u} - \mu \right| > \frac{\epsilon M'}{|T|} \cdot \mu \right]
\]

Using the Chernoff bound of Lemma 2.7

\[
P\left[ \left| \sum_{x,u} X_{x,u} - \mu \right| > \delta \mu \right] < 2e^{-\frac{\delta^2 \mu}{3}} = 2^{-\gamma(\mu \delta^2)}
\]

for a constant \( \gamma \), gives

\[
2^{-\gamma(\mu \delta^2)} = 2^{-\gamma \left( \frac{DK'}{M'} \left( \frac{\epsilon M'}{|T|} \right)^2 \right)} = 2^{-\gamma DK' \epsilon^2 \frac{|T|}{M'}} \leq 2^{-\gamma DK' \epsilon^2},
\]

where at the last step, we used the fact that \( \frac{|T|}{M'} \leq 1 \).

For a fixed \( k' \) and a fixed \( k'-source X \) we get the union bound for the probability that \( C \) is not an extracting conductor:

\[
P \left[ \text{C not an extracting conductor} \right] \leq \sum_{T \subseteq \{0, 1\}^{m'}} P[\mathcal{B}] \leq 2^{M'} \cdot 2^{-\gamma DK' \epsilon^2}
\]

\[
= 2^{K'De^2 A} 2^{-\gamma DK' \epsilon^2} = 2^{-\gamma' DK' \epsilon^2}
\]

for the constant \( \gamma' = \gamma - A \).

And for all possible flat \( k'-sources X \) with \( k' \in [k_{\min}, k_{\max}] \) and thus, \( |X| \in [K_{\min} = 2^{k_{\min}}, K_{\max} = 2^{k_{\max}}] \), we get
\[
P[C \text{ not an extracting conductor}] \leq \sum_{i=K_{\min}}^{K_{\max}} \left( \binom{N}{i} \cdot 2^{-\gamma D \epsilon^2} \right)
\leq \sum_{i=K_{\min}}^{K_{\max}} \left( \frac{N e}{i} \cdot 2^{-\gamma D \epsilon^2} \right)
\leq \sum_{i=K_{\min}}^{K_{\max}} \left( \frac{N e}{K_{\min}} \cdot 2^{-\gamma D \epsilon^2} \right)^i
\leq \sum_{i=K_{\min}}^{K_{\max}} \left( \frac{N e}{K_{\min}} \cdot 2^{-\gamma D \epsilon^2} \right)^i \leq \frac{1}{2}
\]

where at Step 1 we used the upper bound \( \binom{N}{i} \leq \left( \frac{N e}{i} \right)^i \) according to Lemma 2.19.

As in Section 3.5.1, we show the following Requirement I which implies the requirement of
\( P[C \text{ not an extracting conductor}] < 1/2 : \)
\[
\frac{N e}{K^d} \cdot 2^{-\gamma^2 D} \leq \frac{1}{4}
\]

For Requirement I, we have
\[
\frac{N e}{K_{\min}} \cdot 2^{-\gamma^2 D} \leq \frac{1}{4}
\]
\[
\log \left( \frac{N e}{K_{\min}} \right) - \gamma^2 D < -2
\]
\[
n - k_{\min} + \log e + 1 < \gamma^2 D
\]
\[
\frac{1}{\gamma}(n - k_{\min}) + O(1) < \epsilon^2 D
\]
\[
\log(n - k_{\min}) + O(1) < d + 2 \log \epsilon
\]
\[
\log(n - k_{\min}) + 2 \log(1/\epsilon) + O(1) < d
\]

which is fulfilled by our choice of \( d = \log(n - k_{\min}) + 2 \log(1/\epsilon) + O(1) \). Hence, Requirement I is fulfilled and we get for \( P[C \text{ not an extracting conductor}] \):
\[
\sum_{i=K_{\min}}^{K_{\max}} \left( \frac{N e}{K_{\min}} \cdot 2^{-\gamma D \epsilon^2} \right)^i \leq \sum_{i=K_{\min}}^{K_{\max}} \left( \frac{1}{4} \right)^i \leq \sum_{i=1}^{\infty} \left( \frac{1}{4} \right)^i = \frac{1}{3} < \frac{1}{2},
\]

where we used \( \sum_{i=1}^{\infty} \frac{1}{q^n} = \frac{1}{1-q} - 1 \) for a \( 0 < q < 1 \).

For the probability of not being an injective conductor, we instantiate Equation 3.5.2 with \( 2^d = \frac{(n-k)B}{\epsilon^2} \). Thus, we get the second requirement
\[
P[C \text{ not injective}] \leq 2^{2n - \frac{m'(n-k)B}{\epsilon^2}} \leq \frac{1}{2}
\]
3.5. PROBABILITY EXISTENCE PROOFS

Requirement II can be reformed to the requirement
\[ 2n - \frac{m'(n - k)B}{e^2} + 1 < 0. \]

Knowing that \( e^2 \in (0, 1) \), it is sufficient to choose the constant \( B \) such that \( B > \frac{3n}{n-k_{\text{min}}} \) and Requirement II will be fulfilled for all \( n, m' > 0 \).

Putting everything together, we get
\[
\Pr[C \text{ not an injective extracting conductor}] < \sum_{i=1}^{\infty} \left( \frac{1}{4} \right)^i + \frac{1}{2} = \frac{1}{3} + \frac{1}{2} < 1.
\]

We showed that for an output length \( m' \) depending on \( k' \) where \( k' \) is the input min-entropy, we get an injective extracting conductor with \( k_m(k') = m'(k') = m' \) output min-entropy. We can set now the output length for all \( k' \in [k_{\text{min}}, k_{\text{max}}] \) to \( m := m'(k_{\text{max}}) \) because \( m'(k') \leq m'(k_{\text{max}}) \) and when the conductor output contains \( k_m(k') \) min-entropy in the first \( m' \) output bits, it will also contain \( k_m(k') \) min-entropy in the (possibly) longer output length \( m \). In particular, for input min-entropy \( k_{\text{max}} \), we have that the output contains \( k_m(k_{\text{max}}) = m \) min-entropy which fulfills the requirement of being an extracting conductor.

Hence, it follows that there must exist with non-zero probability an injective extracting \( (n, k_{\text{min}}, k_{\text{max}} \times (d) \rightarrow (m, k_m(k'))) \) conductor \( C \) with \( k' \in [k_{\text{min}}, k_{\text{max}}] \), output min-entropy \( k_m(k') = k' + d - \log(1/\epsilon) - c \), seed length \( d = \log(n - k_{\text{min}}) + 2 \log(1/\epsilon) + O(1) \) and output length \( m = k_{\text{max}} + d - \log(1/\epsilon) - c \) for a constant \( c \).

3.5.3 Existence of Injective Expanders

In the last two sections, we have non-constructively shown the existence of good conductors in the sense of using a short seed. This conductors are expander graphs with left-degree \( D \in \Theta(n) \) and \( \gamma = (1 - \epsilon) \cdot D \cdot 2^{-\Delta} \) where \( \epsilon \) is the error and \( \Delta \) the entropy loss of the conductors. In this section, we show non-constructively the existence of expander graphs which have a left-degree in \( O(n) \) for an arbitrary expansion factor. Namely, we prove the Theorem 3.32.

**Theorem 3.32.** There exists an injective \((N, K_{\text{min}}, K_{\text{max}}) \times (D) \rightarrow (M) \) expander graph \( G = (V_1, V_2, \mathcal{E}) \) with \( N = 2^n \), \( M = 2^m \) and \( D = \frac{2 \gamma \log e + n}{m - \log K_{\text{max}}} + 3 \) and \( \gamma \leq D \).

**Proof.** We assume that there exists no graph which is an expander graph as described in Theorem 3.32. We construct a random graph \( G \) by setting \( V_1 = \{0, 1\}^n, V_2 = \{0, 1\}^m \) and \( D = \frac{2 \gamma \log e + n}{m - \log K_{\text{max}}} + 3 \) and for every vertex \( v \in V_1 \), we choose independently random \( D \) neighbors in \( V_2 \).

Let \( S \subseteq V_1 \) be a set with \( |S| = K \) for a \( K \in [K_{\text{min}}, K_{\text{max}}] \). Assume \( G \) is not an expander graph with expansion factor \( \gamma \), i.e. there exist a \( S \) with \( \Gamma(S) < \gamma |S| \). It follows that there must exist a set \( T \subseteq V_2 \) with \( |T| = \gamma |S| \) and \( \Gamma(S) \subseteq T \).

For the probability that a neighbor of \( S \) lies in \( T \), we get \( \frac{|T|}{M} = \frac{\gamma |S|}{M} = \frac{2K}{MK_{\text{DK}}} \), where \( M = |V_2| = 2^m \) and the probability of all \( D \cdot K \) neighbors of \( S \) being in \( T \) is \( \left( \frac{\gamma K}{MK_{\text{DK}}} \right)^{DK} \).

In total, the probability that there exists a set \( S \) with \( K_{\text{min}} \leq |S| \leq K_{\text{max}} \) and \( |\Gamma(S)| < \gamma |S| \) is maximal
\[
\Pr[G \text{ is not an expander}] \leq \sum_{K=K_{\text{min}}}^{K_{\text{max}}} \binom{N}{K} \cdot \binom{M}{\gamma K} \cdot \left( \frac{\gamma K}{MK_{\text{DK}}} \right)^{DK}.
\]
With \( \binom{N}{K} \leq N^K \) and \( \binom{M}{\gamma K} \leq \left( \frac{eM}{\gamma K} \right)^\gamma K \) from Lemma 2.19, we get

\[
P[G \text{ is not an expander}] \leq \sum_{K=K_{\text{min}}}^{K_{\text{max}}} N^K \cdot \left( \frac{eM}{\gamma K} \right)^\gamma K \cdot \left( \frac{\gamma K}{M} \right)^{D-\gamma K}
\]

\[
\leq \sum_{K=K_{\text{min}}}^{K_{\text{max}}} N^K e^\gamma K \left( \frac{\gamma K}{M} \right)^{(D-\gamma)K}
\]

\[
\leq \sum_{K=K_{\text{min}}}^{K_{\text{max}}} Ne^\gamma \left( \frac{\gamma K}{M} \right)^{(D-\gamma)K}
\]

\[
\leq \sum_{K=K_{\text{min}}}^{K_{\text{max}}} Ne^\gamma \left( \frac{2\log(\gamma K_{\text{max}})-m}{M} \right)^{(D-\gamma)K}
\]

Inserting \( D = \frac{2+\gamma \log e+n}{m-\log(K_{\text{max}})} + \gamma \) leads to

\[
P[G \text{ is not an expander}] \leq \sum_{K=K_{\text{min}}}^{K_{\text{max}}} Ne^\gamma \left( \frac{2\log(\gamma K_{\text{max}})-m}{M} \right)^{(D-\gamma)K}
\]

\[
= \sum_{K=K_{\text{min}}}^{K_{\text{max}}} \left( \frac{1}{4} \right)^K \leq \sum_{K=1}^{\infty} \left( \frac{1}{4} \right)^K = \frac{1}{3}
\]

Thus, we get \( P[G \text{ is not an expander}] < 1/3 \). Because we are interested in injective expander graphs, it remains to show that the probability of being not injective is smaller than \( 2/3 \).

Replacing \( 2^d \) with \( D \) in Equation (3.5.2), leads to the probability

\[
2^{2n} \cdot 2^m D = \frac{N^2}{M^2} < \frac{2}{3}
\]

where we set \( 2^n = N \) and \( 2^m = M \).

We assume that \( m \in O(n) \) and thus, there must exist a constant \( \alpha \in (\frac{1}{n}, 1) \) such that \( m = \alpha n \).

We get

\[
\frac{N^2}{M^2} = 2^{n(2-\alpha D)} < 2^{n(2-3\alpha)} \leq 2^{-n} \leq \frac{1}{2} < \frac{2}{3},
\]

where we used \( D > 3 \) from the preconditions and assumed an \( n > 0 \).

Putting everything together leads to

\[
P[G \text{ is not an injective expander}] < \frac{1}{2} + \frac{2}{3} = 1
\]
Hence, with non-zero probability there exist an injective \((N, K_{\min}, K_{\max}) \times (D) \xrightarrow{\gamma} (M)\) expander graph \(G\) with left-degree \(D = \frac{2+\gamma \log e + n}{m - \log(K_{\max} \gamma)} + \gamma\) for an arbitrary \(\gamma \leq D\).

3.6 Application of Expander Graphs

Expander Graphs have many applications such as interpreting them as \textit{input-restricting functions} done in [MT07]. We already gave an overview of this application in Section 1.2. In this section, We show how to interpret an expander graph as an input-restricting function family. Further, to be interesting for [MT07], the expander graph should be constructably polynomial-time and highly unbalanced, i.e. explicit expander graphs with \(|V_1| = 2^n \gg |V_2| = 2^m\) and most importantly for the application in [MT07], the graph should have a left-degree polynomial in \(n\).

First, we recall the definition of \(\mathcal{S}_n\) being an input-restricting function family.

\textbf{Definition 3.33} (input-restricting function). Let \(\epsilon = \epsilon(n) \in (0,1), r = r(n), \delta = \delta(n), m = m(n)\) be functions of \(n\) and let \(n > m\), then a family \(\mathcal{S}_n\) of functions \(E_1, \ldots, E_r : \{0,1\}^n \rightarrow \{0,1\}^m\) is called \((n, \delta, \epsilon, \text{input restricting})\) if it satisfies the following two properties:

\textbf{Injective:} \(\forall x \neq x' \in \{0,1\}^n, \exists i \in 1, \ldots, r\) such that \(E_i(x) \neq E_i(x')\).

\textbf{Input-Restricting:} For all subsets \(\mathcal{M}_1, \ldots, \mathcal{M}_r \subset \{0,1\}^m\) such that \(|\mathcal{M}_1| + \ldots + |\mathcal{M}_r| \leq 2^{n(1-\epsilon)}\), we have

\[
\left| x \in \{0,1\}^n \mid E_i(x) \in \mathcal{M}_i \text{ for all } i = 1, \ldots, r \right| \leq \delta \cdot (|\mathcal{M}_1| + \ldots + |\mathcal{M}_r|).
\]

\(\mathcal{S}_n\) is called \textit{explicit} if \(r(n)\) is polynomial in \(n\) and if \(E_i(\cdot)\) can be computed in \(\text{poly}(n)\) time.

It is clear that \(\delta \geq 1/r\) must hold. Furthermore, we are interested in \(\mathcal{S}_n\) being explicit. We show now how to interpret an expander graph to get \(\mathcal{S}_n\).

Let us assume that we have an explicit (injective) \((2^n, 0, K) \times (D) \xrightarrow{\gamma} (2^m)\) expander graph \(G = (V_1, V_2, E)\) with left-degree \(D\) in \(\text{poly}(n)\). As already mentioned in Lemma 3.3, such an expander graph can be interpreted as a function \(F_G : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m\) which calculates the \(i\)th neighbor for a vertex in \(V_1\). We define \(E_i(x) := F_G(x, i)\) being the \(i\)th neighbor of \(x\) in the expander graph \(G\), i.e. \(E_i : \{0,1\}^n \rightarrow \{0,1\}^m\) for \(i = 1, \ldots, D\). Let \(x^{(i)} = x_{i+1}, \ldots, x_{i+m}\) be the \(i\)th \(m\)-bit substring of \(x\) where extra zeros are appended to \(x\) to make it a multiple of \(m\). Then, if \(G\) is not injective, we define additionally \(E_{D+1}, \ldots, E_{D+[n/m]}\) as \(E_{D+i}(x) = x^{(i)}\) for \(i = 1, \ldots, [n/m]\). We get \(\mathcal{S}_n = \{E_1, \ldots, E_r\}\), where \(r := D\) in the injective case and \(r := D + [n/m]\) if \(G\) is not injective.

We show now that this transformation of an expander graph to a family of functions \(\mathcal{S}_n = E_1, \ldots, E_r\) gives actually an input-restricting function family.

\(\mathcal{S}_n\) is \textit{explicit}: Because \(G\) is an explicit expander graph with polynomially-bounded left-degree \(D\), we have that \(\mathcal{S}_n\) is also explicit.

\(\mathcal{S}_n\) is \textit{injective}: If the expander graph \(G\) is injective, it trivially follows that \(\mathcal{S}_n\) must be injective, too. For the case, where \(G\) is not injective, we assume without loss of generality that \(n\) is a multiple of \(m\). Then, we have for all \(x \in \{0,1\}^n\) that \(x = E_{D+1}(x)[\ldots]|E_{D+[n/m]}(x)\). Let now \(x' \in \{0,1\}^n\) with \(x' \neq x\) and hence, there must exist an \(i \in 1, \ldots, [n/m]\) such that \(E_{D+i}(x) \neq E_{D+i}(x')\). Thus, \(\mathcal{S}_n\) must be injective.
CHAPTER 3. EXPANDERS AND CONDUCTORS

\( \mathcal{F}_n \) is input-restricting: We show this by contradiction. Let \( \mathcal{M}_1, ..., \mathcal{M}_r \subseteq \{0,1\}^m \) be \( r \) subsets with \( |\mathcal{M}_1| + ... + |\mathcal{M}_r| \leq 2^{m(1-\epsilon)} \). Furthermore, let
\[
\mathcal{X} := \left\{ x \in \{0,1\}^n \mid E_i(x) \in \mathcal{M}_i \forall i = 1, ..., r \right\}
\]
such that it would not fulfill the requirement of input-restricting, i.e.:
\[
|\mathcal{X}| > \delta \cdot (|\mathcal{M}_1| + ... + |\mathcal{M}_r|) \tag{3.6.1}
\]
and let \( \mathcal{M} := \bigcup_{i=1}^r \mathcal{M}_i \). According to the definition of the \( E_i(\cdot) \) for \( i = 1, ..., D \), we have \( \Gamma(\mathcal{X}) \subseteq \mathcal{M} \) and therefore, \( |\Gamma(\mathcal{X})| \leq |\mathcal{M}| \). We distinguish now two cases:

**Case 1.** \( |\mathcal{X}| \leq K \).

With \( \gamma = 1/\delta \), we get the following contradiction:
\[
|\mathcal{M}| \geq |\Gamma(\mathcal{X})| \geq \frac{1}{\delta} \cdot |\mathcal{X}|
\]
\[
> \frac{1}{\delta} \cdot \delta \cdot (|\mathcal{M}_1| + ... + |\mathcal{M}_r|)
\]
\[
\geq |\mathcal{M}| \gamma \tag{3.6.2}
\]
where we used Equation (3.6.1) at Step (3.6.2).

**Case 2.** \( |\mathcal{X}| > K \).

We use the definition of \( \epsilon \) and get
\[
2^{m(1-\epsilon)} < 2^{m(1-(1-\log(\gamma K)/m))} = \gamma K.
\]
Let \( \mathcal{X}' \subset \mathcal{X} \) such that \( |\mathcal{X}'| = K \). This gives
\[
\gamma K > 2^{m(1-\epsilon)} \geq |\mathcal{M}_1| + ... + |\mathcal{M}_r| \geq |\mathcal{M}| \geq |\Gamma(\mathcal{X})| \geq |\Gamma(\mathcal{X}')| \geq \gamma \cdot |\mathcal{X}'| = \gamma K
\]
which is again a contradiction.

Therefore, \( |\mathcal{X}| \leq \delta \cdot (|\mathcal{M}_1| + ... + |\mathcal{M}_r|) \) must hold and not Equation (3.6.1) which leads to \( \mathcal{F}_n \) being input-restricting. Hence, \( \mathcal{F}_n \) satisfies all requirements needed to be an input-restricting function family and we state the result in the next lemma.

**Lemma 3.34.** Let \( n \) be such that \( n > m \). Assume that there exists an explicit \( (2^n, K) \times (D) \to (2^m, \gamma) \) expander graph \( G = (V_1, V_2, \mathcal{E}) \) with \( \text{poly}(n) \)-left-degree \( D \) where \( V_1 = \{0,1\}^n \) and \( V_2 = \{0,1\}^m \). Then, for all \( \epsilon > 0 \) such that \( \epsilon > 1 - \frac{\log(K\gamma)}{m} \) for \( m \) large enough, there exists an explicit \( (n, \delta, \epsilon) \)-input-restricting family of functions with \( \delta = 1/\gamma \) and cardinality \( r := D + \lceil n/m \rceil \). If \( G \) is injective then there exists the same input-restricting family of functions but with smaller cardinality \( r := D \).

For the application described in Section 1.2 we want \( \epsilon \) being as small as possible such that the number of allowed queries \( 2^{(1-\epsilon)m} \) exceed the so called birthday barrier \( O(2^{n/2}) \). Hence, together with \( m \in O(n) \), we require a big upper bound \( K \) of the order \( 2^{\Theta(n)} \).

In our derivation of input-restricting functions, we assumed the existence of an explicit \( (N, 0, K) \times (D) \to (M) \) expander graph with polynomially-bounded left-degree and \( K \in 2^{\Theta(n)} \). The purpose of the next Chapters 4 and 5 will be to find an explicit construction of such an expander graph and in Section 5.4 we will actually present a construction which leads to an expander graph satisfying the requirements stated in Lemma 3.34 in computationally-theoretic terms.
3.7 Notations Used in the Literature

In Section 3.1 and 3.2 we introduced a generalized notion of expander graphs and conductors. We give here a short reference to the notations and notions of different combinatorial functions and expander graphs used in the literature which are special cases of our generalized conductors, and show how they map our definition of conductors and expander graphs. We also give citations to publications as an example for the usage of the described notions.

In the literature, the notion of a \((K, \gamma)\)-expander graph \(G = (V_1, V_2, E)\) with left-degree \(D\) is often used \([TUZ01]\). In our framework, \(G\) is a special case of expander graphs, namely, an \((\lvert V_1 \rvert, 0, K) \times (D) \rightarrow (\lvert V_2 \rvert)\) expander graph. Note that outside my thesis, the lower bound for the size of the set \(\mathcal{X}\) is always assumed to be zero.

In the setting of conductors, a \((k, \alpha, \epsilon)\)-conductor \(C : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m\) as described in \([CRVW02, MT07]\) would be in our generalized framework an \((n, 0, k) \times (d) \rightarrow_{\epsilon} (m, k' \mapsto k' + \alpha)\) generalized conductor and the notion of an \((n, m, d, k, \epsilon)\)-extractor as defined in \([Tre98, TUZ01]\], is an extracting \((n, k, k) \times (d) \rightarrow_{\epsilon} (m, m)\) generalized conductor. For the case, where we have a fixed input min-entropy as for extractors, but have \(k_m(k') = k_m\) for a value \(k_m\) not necessarily equals to the output length, the notion of an \((n, k) \rightarrow_{\epsilon} (m, k_m)\) condenser is known \([TUZ01]\).
4 Basic Constructions

The purpose of this chapter is to introduce some basic constructions of conductors which we will use later in Chapter 5 to construct new conductors by using the composition theorems of Section 3.3, and in particular, to construct an expander graph needed for the application described in Section 1.2. In Section 4.1, we introduce a construction due to Trevisan\textsuperscript{1} [Tre98]. Then, in Section 4.2, we describe an extracting conductor which works for small min-entropies and which uses hash functions as basic elements. Finally, we explain a construction of a lossless conductor in Section 4.3.

4.1 Trevisan’s Extracting Conductor

In this section, we present a construction of an explicit strong extracting conductor due to Trevisan [Tre98]. This conductor is of special interest, because it is often used as a subfunction to construct other conductors.

The main idea of Trevisan’s construction is to use a special kind of a pseudo-random generator (PRG) due to Nisan-Wigderson [NW94] which relies on a hard to compute function. This PRG can be used to obtain a strong injective extracting conductor.

4.1.1 Nisan-Wigderson Pseudo-Random Generator

A pseudo-random generator (PRG) is a function for which there is no small circuit distinguishing the PRG output efficiently from a uniformly distributed string. In [NW94], it was shown that if we have a function $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$ which is on average-case hard to compute, then we can construct a PRG with the help of black-box calls to the function $f$. With “hard to compute”, we mean that for all algorithms $A$ with circuit complexity at most $2^{\gamma \ell}$ for a constant $\gamma > 0$, we have $\Pr[A(x) = f(x)] < 1/2 + \text{negligible}.$

We will present an example for such a hard function $f$ in Section 4.1.2, where we use an error-correcting (EC) encoding of an $n$-bit string with enough min-entropy and interpret the encoding as a truth table of a boolean function. But for this section, we just assume the existence of such a hard function $f$ and show how to construct the NW generator.

The NW generator $NW_{\mathcal{S}, f} : \{0, 1\}^d \rightarrow \{0, 1\}^m$ is defined as

$$NW_{\mathcal{S}, f}(y) := f(y|S_1) \cdots f(y|S_m),$$

where $\mathcal{S} = S_1, ..., S_m$ is a collection of subsets of $[d]$ each of size $\ell$ and $y|S_i$ is the string in $\{0, 1\}^\ell$ obtained by projecting $y$ onto the coordinates specified by $S_i$.

To ensure that the $y|S_i$ look like independently random $\ell$-bit strings, the set $\mathcal{S}$ has to be a weak $(m, d, \ell, \rho)$-design as we will show later.

\textsuperscript{1}which is one of the most famous extractor constructions, or in our setting an extracting conductor construction
CHAPTER 4. BASIC CONSTRUCTIONS

Note. The original NW generator used \((m,d,\ell,a)\)-designs but Vadhan [Vad98] showed that a weaker notion of designs also meets the requirements, namely so called weak designs. We use a stronger notion of weak designs than [Vad98], which is required later to get a conductor with \(k_{\text{min}} \neq k_{\text{max}}\) with Trevisan’s construction. We go more into detail in Section 4.1.2.

Definition 4.1 (weak design). A family of sets \(\mathcal{S} = S_1, \ldots, S_m \subset [d]\) is a weak \((m,d,\ell,\rho)\)-design if

1. For all \(i, |S_i| = \ell\)
2. For all \(i, \sum_{j<i} 2^{|S_i \cap S_j|} \leq \rho \cdot (i - 1)\).

Let now \(f\) be a bad function for NW, i.e. there exists a distinguisher \(D\) which can distinguish the output of the NW pseudo-random generator from a random string with success probability of at least \(1/2 + \epsilon\) and hence, NW is not a pseudo-random generator as we prove below. In this case, \(D\) can be used to calculate a function \(g : \{0,1\}^\ell \rightarrow \{0,1\}\) which has the same output as the hard function \(f\) for a fraction of \(1/2 + \epsilon/m\) inputs (we say \(g\) approximates \(f\) within \(1/2 + \epsilon/m\)). If we use a weak \((m,d,\ell,\rho)\)-design \(\mathcal{S}\), the number of needed functions \(g\) to approximate all bad functions \(f\), given a distinguisher \(D\), is upper bounded by \(2^{1+\log m + \rho \cdot (m-1)}\). Additionally, we show that even revealing the random \(d\)-bit string used for NW does not compromise the security because the hard function \(f\) is not efficiently computable and hence, one cannot use the seed for distinguishing the NW output from a truly random bit string. In particular, we prove the following lemma.

Lemma 4.2. Let \(\mathcal{S}\) be a weak \((m,d,\ell,\rho)\)-design, and \(D : \{0,1\}^m \rightarrow \{0,1\}\) a distinguisher. Then there exists a family \(G_D\) of at most \(2^{1+\log m + \rho \cdot (m-1)}\) functions such that for every function \(f : \{0,1\}^\ell \rightarrow \{0,1\}\) satisfying

\[
|P[D(NW_{\mathcal{S}}(f)(U_d), U_d) = 1] - P[D(U_m, U_d) = 1]| \geq \epsilon
\]

there exists a function \(g : \{0,1\}^\ell \rightarrow \{0,1\}, g \in G_D\) such that \(g(\cdot)\) approximates \(f(\cdot)\) within \(1/2 + \epsilon/m\).

In the proof of Lemma 4.2 we show additionally that every such function \(g\) can be described by using at most \(1 + \log m + \rho \cdot (m-1)\) bits, given \(D\) as a circuit. We call this description bit string for function \(g\) the advice string. Hence, given the advice string and the distinguisher \(D\), we can construct a reconstruction function \(g\) for the function \(f\). We will talk more about this when introducing a strong condensing conductor in Section 4.3.

Proof of Lemma 4.2. Our proof method is based on [Tre98, Vad98] but is extended to the case where we append the used random bit string \(y\) to the output of the NW pseudo-random generator. For the proof, we use the so called hybrid argument. If the distinguisher \(D\) can distinguish \((NW_{\mathcal{S}}(f)(y), y)\) from \((U_m, U_d)\) with success of at least \(\epsilon\), then there must be at least one position in the output string where this distinction is noticeable otherwise \(D\) cannot find any difference between the two output strings and therefore, does not distinguish \((NW_{\mathcal{S}}(f)(y), y)\) from \((U_m, U_d)\). Let \(H_0, \ldots, H_m\) be \(m + 1\) distributions with \(H_i := (v_1 \ldots v_i r_{i+1} \ldots r_m y_1 \ldots y_d)\) where \(v \equiv NW_{\mathcal{S}}(f)(y)\) for a random \(y\) and \(r \in \{0,1\}^m\) be a truly random bit string. It is easy to see that \(H_m\) is \((NW_{\mathcal{S}}(f)(y), y)\) and \(H_0\) is the uniform distribution over \(\{0,1\}^m \times \{0,1\}^d\).
Given the Equation (4.1.1), there must be a bit $b_0 \in \{0, 1\}$ such that

$$P \left[ D(NW_{f}, U_d) \oplus b_0 = 1 \right] - P \left[ D(U_m, U_d) \oplus b_0 = 1 \right] \geq \epsilon.$$

To simplify the notation, we just define a new distinguisher $D'() := D() \oplus b_0$. Note that the value of $b_0$ can be easily found by choosing $b_0 \in \{0, 1\}$ such that

$$P \left[ D(NW_{f}, U_d) \oplus b_0 = 1 \right] - P \left[ D(U_m, U_d) \oplus b_0 = 1 \right]$$

is maximized. Using the distributions $H_i$ and the new distinguisher $D'$ this can be rewritten to

$$\epsilon \leq P \left[ D'(NW_{f}, U_d) = 1 \right] - P \left[ D'(U_m, U_d) = 1 \right] = P \left[ D'(H_m) = 1 \right] - P \left[ D'(H_0) = 1 \right] = \sum_{i=1}^{m} P \left[ D'(H_i) = 1 \right] - P \left[ D'(H_{i-1}) = 1 \right]$$

and hence, there must be an index $i$ with

$$P \left[ D'(H_i) = 1 \right] - P \left[ D'(H_{i-1}) = 1 \right] \geq \epsilon/m$$

and

$$H_i = f(y|S_1) \cdots f(y|S_r)r_{i+1} \cdots r_{m}y_1 \cdots y_d$$

$$H_{i-1} = f(y|S_1) \cdots f(y|S_{r-1})r_{i} \cdots r_{m}y_1 \cdots y_d.$$ We can assume without loss of generality that $S_i = \{1, \ldots, \ell\}$. Let $y := (x, z)$ where $x = y|S_1 \in \{0, 1\}^\ell$ and $z = y_{|d-S_1} \in \{0, 1\}^{d-\ell}$ the not taken bits. Further, let $h_j(x, z) := y|S_j$ for every $j < i$ and $y = (x, z)$. We see that $h_j(x, z)$ depends on $|S_i \cap S_j|$ bits of $x$ and on $\ell - |S_i \cap S_j|$ bits of $z$. Putting everything together and using the fact that $P_X[1] = E[X]$, we have

$$E \left[ D'(f(h_1(x, z)), \ldots, f(h_{i-1}(x, z)), f(x), r_{i+1}, \ldots, r_m, y_1, \ldots, y_d) \right] - E \left[ D'(f(h_1(x, z)), \ldots, f(h_{i-1}(x, z)), f(x), r_{i+1}, \ldots, r_m y_1, \ldots, y_d) \right]$$

$$= E \left[ D'(f(h_1(x, z)), \ldots, f(h_{i-1}(x, z)), f(x), r_{i+1}, \ldots, r_m y_1, \ldots, y_d) \right] - D'(f(h_1(x, z)), \ldots, f(h_{i-1}(x, z)), r_{i+1}, \ldots, r_m y_1, \ldots, y_d) \geq \epsilon/m.$$ Without loss of generality\(^2\), we fix the random $r_{i+1}, \ldots, r_m$ to some values $c_{i+1}, \ldots, c_m$ and $z$ to some value $w$ because $z$ is independent of $x$.

$$E_{r_i, z} \left[ D'(f(h_1(x, w)), \ldots, f(h_{i-1}(x, w)), f(x), c_{i+1}, \ldots, c_m, y_1, \ldots, y_d) \right] - D'(f(h_1(x, w)), \ldots, f(h_{i-1}(x, w)), r_{i+1}, c_{i+1}, \ldots, c_m, y_1, \ldots, y_d) \geq \epsilon/m.$$\(^2\)

\(^2\)using an averaging argument
CHAPTER 4. BASIC CONSTRUCTIONS

We rename $r_i$ to $b$ and define now a new function $F : \{0,1\}^{t+1} \rightarrow \{0,1\}^m$ with $F(x,b) = f(h_1(x,w)), \ldots, f(h_{i-1}(x,w)), b, c_{i+1}, \ldots, c_m, y_1, \ldots, y_d$ for a fixed $w$. Note that the bits $y_1, \ldots, y_d$ are fully described by $x$ and $w$ and we can therefore omit $y_1, \ldots, y_d$ as inputs for the function $F$. Inserting in above equation gives

$$P_{x,b} [D'(F(x,f(x))) = 1] - P_{x,b} [D'(F(x,b)) = 1] > \epsilon /m.$$ 

Therefore, function $F$ and $D'$ can be used to distinguish the pair $(x,f(x))$ from $U_{t+1}$. We will show how one can use $F$ and $D'$ to construct a function $g(\cdot)$ which agrees with $f(\cdot)$ on a fraction of $1/2 + \epsilon /m$ of the domain. Choose $b \in \{0,1\}$ and compute $D'(F(x,b))$. If $D'(F(x,b)) = 1$ then output $g(x) = b$, else output $g(x) = 1 - b$. The probability that $g(x)$ agrees with $f(x)$ is

$$P_{b,x} [g(x) = f(x)] = P_{b,x} [g(x) = f(x)|b = f(x)] P_{b,x} [b = f(x)] + P_{b,x} [g(x) = f(x)|b \neq f(x)] P_{b,x} [b \neq f(x)]$$

$$= \frac{1}{2} P_{b,x} [D'(F(x,b)) = 1|b = f(x)] + \frac{1}{2} P_{b,x} [D'(F(x,b)) = 0|b \neq f(x)]$$

$$= \frac{1}{2} \left( P_{b,x} [D'(F(x,b)) = 1|b = f(x)] - P_{b,x} [D'(F(x,b)) = 1|b \neq f(x)] \right)$$

$$\geq \frac{1}{2} + \frac{\epsilon}{m}.$$ 

Putting everything together, we see that for describing function $F$ we need log $m$ bits to specify $i$, we need one bit for $b$ and for every $j < i$ and $x$ we have to describe $f(h_j(x,w))$ and for $j > i$ we have to describe $c_j$. Note that for fixed $w$, the outcome $h_j(x,w)$ for $j < i$ depends only on $|S_i \cap S_j|$ bits of $x$. Hence, although $f$ cannot be efficiently computed, we can just give all possible outputs for inputs $h_j(x,w)$ to describe function $g$, which are $2^{|S_i \cap S_j|}$ values of $f$ for every $j < i$. Overall, $g(\cdot)$ can be described by using $1 + \log m + \sum_{j < i} 2^{|S_i \cap S_j|} + (m - i)$ $\leq 1 + \log m + \rho \cdot (m - 1)$ bits. Thus, there are at most $2^{1+\log m + \rho \cdot (m-1)}$ possible functions $g(\cdot)$ and therefore $|G_D| \leq 2^{1+\log m + \rho \cdot (m-1)}$.

4.1.2 Making the NW Generator an Injective Extracting Conductor

In this section, we show how to get an injective strong extracting conductor from the NW generator.

The NW generator extends $d$ truly random bits to $m$ almost truly random bits. To achieve this extension, one needs a hard predicate function $f$. When we declare an $(n,k_{\min},k_{\max}) \times (d) \rightarrow (m,k_m(k'))$ extracting conductor $Ext$, we do not just have the $d$ random bits as input, but also $n$ bits from a $k'$-source where $k' \in [k_{\min},k_{\max}]$. The idea is to use this $n$ additional bits for defining a hard predicate function $f$.

In particular, one uses an error-correcting code $EC : \{0,1\}^n \rightarrow \{0,1\}^\ell$ code and computes the encoding $\bar{x} = EC(x)$ of the $n$ input bits $x$. Further, we interpret the encoding $\bar{x}$ as a truth table of a predicate function $\bar{x} : \{0,1\}^\ell \rightarrow \{0,1\}$ with $\ell = \log n$ and we will show that function $\bar{x}$ is on average a hard function when the $n$-bit string is drawn from a distribution with sufficient min-entropy. To illustrate the idea, let $EC(x) = \bar{x} = b_1 b_2 \cdots b_n$ where $b_i$ is the $i$-th bit of the EC encoding $\bar{x}$. Then we can interpret $b_1 b_2 \cdots b_n$ as a function table as Table 4.1.
Lemma 4.4. Therefore compromise the randomness of the NW generator output.

In particular, every function $g \in G_D$ such that $g(\cdot)$ approximates $\bar{x}(\cdot)$ within $1/2 + \epsilon/m = 1/2 + \delta$. Additionally, we know that there are at most $|G_D| \leq 2^1 + \log m + \rho(m-1)$ such functions $g_i$’s. In particular, every function $g \in G_D$ can be an approximation of at most $(m/\epsilon)^2$ different functions $\bar{x}(\cdot)$, because of the EC encoding according to Lemma 4.3 and hence, every Hamming ball with relative distance $1 + \epsilon/m$ to $g$ contains at most $(m/\epsilon)^2$ encodings $\bar{x}$ of different $x$’s. Overall, $|G_D| \cdot (m/\epsilon)^2 = 2^{1+\log m + \rho(m-1)} \cdot (m/\epsilon)^2$ is an upper bound on the number of bad strings $x$ for which Expression (4.1.2) can occur.

Lemma 4.3. [Tre98] For every $n$ and $0 \leq \delta < 1/2$, there is a polynomial-time computable encoding $EC_\delta : \{0,1\}^n \rightarrow \{0,1\}^n$ where $\bar{n} = poly(n,1/\delta)$ such that every ball of Hamming radius $(1/2 - \delta)n$ in $\{0,1\}^n$ contains at most $1/\delta^2$ codewords. Furthermore, $\bar{n}$ can be assumed to be a power of $2$.

For our conductor, we use such an error-correcting encoding with $\delta = \epsilon/m$, we will show later in the proof of Theorem 4.5 why. We state now that if $\delta = \epsilon/m$, there are only few $x \in \{0,1\}^n$ where its encoding $\bar{x}$ is not a truth table of a hard predicate function and would therefore compromise the randomness of the NW generator output.

Lemma 4.4. Let $EC : \{0,1\}^n \rightarrow \{0,1\}^n$ be an error-correcting code fulfilling Lemma 4.3 with $\delta = \epsilon/m$. Then for $Ext_{\gamma,EC}(x,y) = NW_{\gamma,\bar{x}}(y) = \bar{x}(y|S_1) \cdots \bar{x}(y|S_m)$ with $\bar{x} = EC(x)$ and for every distinguisher $D : \{0,1\}^m \rightarrow \{0,1\}$, there are at most $2^{1+\log m + \rho(m-1)} \cdot (\epsilon/m)^2$ strings $x \in \{0,1\}^n$ such that

$$|P[D(Ext_{\gamma,EC}(x,U_d),U_d) = 1] - P[D(U_m,U_d) = 1]| \geq \epsilon. \quad (4.1.2)$$

Proof. Figure 4.1 is a helpful illustration for this proof. If $x$ is a bad string, i.e. it is such that Equation (4.1.2) holds, then from Lemma 4.2 we know that there exists a function $g_i : \{0,1\}^\ell \rightarrow \{0,1\}$ in $G_D$ such that $g(\cdot)$ approximates $\bar{x}(\cdot)$ within $1/2 + \epsilon/m = 1/2 + \delta$. Additionally, we know that there are at most $|G_D| \leq 2^{1+\log m + \rho(m-1)}$ such functions $g_i$’s. In particular, every function $g \in G_D$ can be an approximation of at most $(m/\epsilon)^2$ different functions $\bar{x}(\cdot)$, because of the EC encoding according to Lemma 4.3 and hence, every Hamming ball with relative distance $1 + \epsilon/m$ to $g$ contains at most $(m/\epsilon)^2$ encodings $\bar{x}$ of different $x$’s. Overall, $|G_D| \cdot (m/\epsilon)^2 = 2^{1+\log m + \rho(m-1)} \cdot (m/\epsilon)^2$ is an upper bound on the number of bad strings $x$ for which Expression (4.1.2) can occur.

\[
\begin{array}{c|c}
\text{all values } \in \{0,1\}^\ell & \text{output} \\
00...00 & b_1 \\
00...01 & b_2 \\
\ldots & \ldots \\
11...10 & b_{n-1} \\
11...11 & b_n \\
\end{array}
\]

Table 4.1: Function table of a function $\{0,1\}^\ell \rightarrow \{0,1\}$
CHAPTER 4. BASIC CONSTRUCTIONS

Note. If we did not use error-correcting codes, then there would be too many functions with relative distance \( t = 1/2 + \epsilon/m \) to \( \bar{x} \). Let \( w = 1 + \log m + \rho(m-1) \) then the number of strings with Hamming distance at most \( t \cdot w \) is:

\[
\sum_{i=0}^{tw} \binom{w}{tw} \approx 2^{w-h(t)} \approx 2^w = 2^{1+\log m+\rho(m-1)}
\]

In the first step we used the upper bound from Lemma 2.20 and in the second step the fact that \( t \) is about \( 1/2 \) and for that the binomial entropy function is 1. Overall, we would have \( 2^{1+\log m+\rho(m-1)} \) strings \( x \) satisfying Expression (4.1.2) which are too many.

**Theorem 4.5.** Let \( n > m \), and \( \epsilon > 0 \) be a constant. If \( \mathcal{J} \) is a weak \((m,d,\ell,\rho)\)-design for \( \rho = (k_{\max} - 3\log(m/\epsilon) - 1)/m \), and \( EC \) from Lemma 4.3 with \( \delta = \epsilon/m \) and \( \ell = \log(n) \), then \( Ext_{\mathcal{J},EC} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m \) is an injective strong extracting \((n,0,k_{\max}) \times (d) \to 2\epsilon \) \((m,k_m(k'))\) conductor with \( k_m(k') = (k' - 3\log(m/\epsilon) - 1)/\rho \) and \( k' \in [0,k_{\max}] \). If \( \mathcal{J} \) can be constructed in \( \text{poly}(m,d) \) time then \( Ext_{\mathcal{J},EC} \) is efficient.

**Proof.** As in the probabilistic proof of Section 3.5.2, we fix the input min-entropy to a value \( k' \in [0,k_{\max}] \) and denote the function \( m' := m'(k') = k_m(k') \) which does describe the first \( m'(k') \) output bits. Note that \( m = m'(k_{\max}) \) and with the weak \((m,d,\ell,\rho)\)-design we have for every \( S_i \)

\[
\sum_{j<i} 2^{S_i \cap S_j} \leq \rho \cdot (i-1).
\]

Hence, we can just ignore the last \( m - m' \) subsets of \( \mathcal{J} \) and get a new weak \((m',d,\ell,\rho)\)-design \( \mathcal{J}' = S_1, ..., S_{m'} \).

We show now that for input min-entropy \( k' \) the output is \( 2\epsilon \)-close to \( Y \circ U_d \), where \( Y \) is a \( m' \)-source.

Redoing the proof of Lemma 4.2 and 4.4 for the input-min-entropy \( k' \) and output length \( m' \), we see that there are at most \( 2^{1+\rho(m'-1)+\log(m')}+2\log(m/\epsilon) \) possible bad strings \( x \) for which condition

\[
|P[D(Ext'_{\mathcal{J},EC}(x,U_d),U_d) = 1] - P[D(U_{m'},U_d) = 1]| > \epsilon
\]
4.1. TREVISAN’S EXTRACTING CONDUCTOR

holds, where we denote with $Ext_{y}$ the first $m'$ output bits of $Ext_{y}$. Because $x$ is selected from a $k'$-source, the probability of each string is at most $2^{-k'}$ to be selected. In total, the probability to select a bad string $x$ is

$$2^{1+\rho(m'-1)+\log(m')/\epsilon} \cdot 2^{-k'} \leq 2 \cdot m^2/\epsilon \cdot m' \cdot 2^m \cdot 2^{-k'}$$

$$= 2 \cdot m^2/\epsilon \cdot m' \cdot 2^{k'-\log(m/\epsilon)-1} \cdot 2^{-k'}$$

$$\leq 2 \cdot m^3/\epsilon \cdot 1/2 \cdot 3/m^3$$

(1)

where at Step (1) we used $m' = (k' - 3 \log(m/\epsilon) - 1)/\rho$ and at Step (2) we used $m' \leq m$.

Then we have

$$|P[D(Ext_{y,EC}(X,U_d), U_d) = 1] - P[D(U_{m'}, U_d) = 1]|$$

$$\leq E_{x \in X} \left| P[D(Ext_{y,EC}(x,U_d), U_d) = 1] - P[D(U_{m'}, U_d) = 1] \right|$$

$$= \sum_{x \in B} P[X = x] \cdot |P[D(Ext_{y,EC}(x,U_d), U_d) = 1] - P[D(U_{m'}, U_d) = 1]|$$

(2)

$$+ \sum_{x \notin B} P[X = x] \cdot |P[D(Ext_{y,EC}(x,U_d), U_d) = 1] - P[D(U_{m'}, U_d) = 1]|$$

$$\leq \epsilon + \epsilon$$

$$= 2\epsilon$$

(3)

where (2) is an application of the triangle inequality and at (3) we used the upper bound 1 for the expressions

$$|P[D(Ext_{y,EC}(x,U_d), U_d) = 1] - P[D(U_{m'}, U_d) = 1]| \quad \text{if} \ x \in B$$

and

$$P[X = x] \cdot |x \notin B].$$

$Ext_{y,EC}(x,y)$ is therefore $2\epsilon$-close to $Y \circ U_d$ for a $k_m(k')$-source $Y$.

It remains to show that $Ext_{y}(x,y)$ is an injective conductor. Recall that for being injective, the following must be true:

$$\forall x': x \neq 0 \cdot 1^n : \exists y \in \{0,1\}^d : Ext_{y,EC}(x,y) \neq Ext_{y,EC}(x',y).$$

We show the injectivity by contradiction: We assume that

$$\exists x' \neq x \in \{0,1\}^n : \forall y \in \{0,1\}^d : Ext_{y,EC}(x,y) = Ext_{y,EC}(x',y).$$

This would mean that:

$$\bar{y}(y_{S_1}) \cdots \bar{y}(y_{S_m}) = \bar{y}'(y_{S_1}) \cdots \bar{y}'(y_{S_m})$$

We fix an $i \in [1,\ldots,m]$. Then we have that

$$|\{y_{S_i} \mid \text{for an } i \text{ and } \forall y\}| = 2^{|S_i|} = 2^i.$$
Because \( \bar{x}(\cdot) \) (and \( \bar{x}'(\cdot) \)) is a function from \( \{0,1\}^d \) to \( \{0,1\} \), this means that \( \bar{x} \) and \( \bar{x}' \) are equal for all possible inputs and in particular \( \bar{x} = EC(x) = \bar{x}' = EC(x') \). This contradicts the property of the EC encoding because \( x \neq x' \). Hence, such a \( x' \) does not exist.

Note that if the weak design \( \mathcal{S} \) can be constructed in \( \text{poly}(m,d) \) time then \( \text{Ext}_{\mathcal{S},EC}(x,y) \) can be computed in \( \text{poly}(n,d) \) time because the EC encoding is efficiently computable (in \( \text{poly}(n) \)). Hence, \( \text{Ext}_{\mathcal{S},EC}(x,y) \) is explicit in this case.

4.1.3 Example Instantiation of Trevisan’s Extracting Conductor

In this section, we present a concrete instantiation of Trevisan’s strong extracting conductor and calculate its parameters. We will set \( m := k_{\text{max}}/2 \), hence the conductor we will get, will extract half of the maximal input min-entropy and preserves the randomness of the seed. With this setting, we get

\[
\rho = 2 \cdot \left( k_{\text{max}} - 3 \log (k_{\text{max}}/(2\epsilon)) - 1 \right)/k_{\text{max}} = 2 - \frac{6 \log (k_{\text{max}}/\epsilon) - 4}{k_{\text{max}}}
\]

which has to be at least 1. Thus, we choose \( \epsilon \) such that

\[
\frac{6 \log (k_{\text{max}}/\epsilon) - 4}{k_{\text{max}}} < 1
\]

which is achieved by the choice

\[
\epsilon > 2^{\log (k_{\text{max}}) - k_{\text{max}}/2 + 2/3}.
\]

Example for an EC Encoding. We use the notation \([N,K,D]_Q\) for linear error-correcting codes where \( N \) is the output length, \( K \) is the input message length, \( D \) the minimal Hamming distance of the code and \( Q \) the alphabet size. A possible linear error-correcting code fulfilling the requirements of Lemma 4.3 is the concatenation\(^4\) of a Reed-Solomon (RS) code \([p, k, p - (k - 1)]_q\) \(^5\) and a Hadamard (Ha) code \([p, \log p, \frac{p}{2}]_2\). The encoding and decoding of this concatenated code can be done in polynomial time. Let \( p \) be a power of 2 and \( \sigma > 1 \), then we can set \( q := p \), \( k := p/\sigma \) and by the fact that a \([p, k, d]_p\) code is also a \([p, k, d - 1]_p\) code we get the RS code \([p, \frac{p}{2}, \frac{p - 1}{\sigma} \cdot p]_p\) code. Concatenating the RS code with the Ha code leads to the following linear code

\[
\left[ p, \frac{p}{\sigma}, \frac{\sigma - 1}{\sigma} \cdot p \right]_p \cdot \left[ p, \log p, \frac{p}{2} \right]_2 \Rightarrow \left[ p^2, \frac{p}{\sigma} \cdot \log p, \left( 1/2 - \frac{1}{2\sigma} \right) \cdot p^2 \right]_2
\]

with relative Hamming distance \( \frac{1}{2} - \frac{1}{2\sigma} \).

The strings \( x \in \{0,1\}^n \) are used as input for the \([p^2, \frac{p}{\sigma} \cdot \log p, \left( 1/2 - \frac{1}{2\sigma} \right) \cdot p^2]_2 \) code and hence, we have \( n = (p/\sigma) \cdot \log p \) and \( \bar{n} = p^2 \). We see that \( p \leq \sigma n \) and thus, \( \bar{n} \) is smaller than \( (\sigma n)^2 \).

We show that this encoding actually has the property required by Lemma 4.3. To achieve this, we use the following bound from [Tre98].

Lemma 4.6. Suppose \( EC \) is an error-correcting code with relative minimum distance \( \geq 1/2 - \beta/2 \). Then every Hamming ball of relative radius \( 1/2 - \sqrt{\beta} \) contains at most \( 1/(3\beta) \) codewords.

\(^4\)by first interpreting the input message as a messages in \( GF(q^k) \) of length \( K \) and applying an \([N,K,D]_{q^k}\) code and second, interpreting the encoding as \( N \) messages of length \( k \) and applying \( N \) times an \([n,k,d]_q\) code on it, we get an \([nN, kK, dD]_q\) code

\(^5\)\( k \leq p \leq q \)
We set \( \beta := 1/\sigma \) and \( \delta := 1/\sqrt{\sigma} \). Applying Lemma 4.6 we get at most \( \sigma/3 \leq \sigma = 1/\delta^2 \) codewords in a Hamming ball of radius \( 1/2 - 1/\sqrt{\sigma} = 1/2 - \delta \) which fulfills the requirements of Lemma 4.3.

For our instantiation of Trevisan’s conductor, we set \( \sigma = m^2/\epsilon^2 \) to get \( \delta \leq m/\epsilon \). We get \( \bar{n} = n^2m^4/\epsilon^2 \) and for the parameter \( \ell \):

\[
\ell = \log(\bar{n}) = \log(n^2m^4/\epsilon^4) = 2\log n + 4\log m + 4\log(1/\epsilon) < 6\log n + 4\log(1/\epsilon),
\]

where at the last step, we assumed that \( n = c \cdot m \) for a constant \( c > 1 \).

**Example for a Weak Design.** The existence of an efficiently constructible weak design with almost ideal \( d \) is stated in [Vad98]:

**Lemma 4.7.** For every \( \ell, m \in \mathbb{N} \) and \( \rho > 1 \), there exists a weak \((m, d, \ell, \rho)\)-design \( \mathcal{S} = S_1, \ldots, S_m \subset [d] \) with

\[
d = \left\lceil \frac{\ell}{\ln \rho} \right\rceil \cdot \ell.
\]

Moreover, such a weak design can be found in \( \text{poly}(m, d) \) time.

We denote a new function \( \psi(k_{\text{max}}, \epsilon) := \left[ \ln(2 - \frac{6\log(k_{\text{max}}/\epsilon)}{k_{\text{max}}}) - 4 \right]^{-1} \) which computes the value \( 1/\ln(\rho) \) depending on \( k_{\text{max}} \) and \( \epsilon \). We will omit the input parameters and write for short-hand just \( \psi \). Hence, we know now that we can set the seed length to

\[
d = \lceil \psi \cdot \ell \rceil \cdot \ell \leq \psi \cdot \ell^2 + \ell.
\]

**Putting Everything Together.** With the chosen EC and weak design, the value of \( d \) is

\[
d \leq \psi \cdot \ell^2 + \ell \leq \psi \cdot (6\log(n) + 4\log(1/\epsilon))^2 + 6\log(n) + 4\log(1/\epsilon)
\]
\[
= 36 \cdot \psi \cdot \log^2(n) + \psi \cdot 48 \cdot \log(n) \cdot \log(1/\epsilon) + 6\log(n) + \psi \cdot 16 \log^2(1/\epsilon) + 4\log(1/\epsilon)
\]
\[
\in O(\log^2(n))
\]

Overall, we get a strong injective extracting \((n, 0, k_{\text{max}}) \times (d) \rightarrow (m, k_m(k'))\) conductor with \( m = k_{\text{max}}/2 \) and \( k_m(k') = (k' - 3\log(m/\epsilon) - 1)/\rho \) for \( k' \in [0, k_{\text{min}}] \) which needs \( O(\log^2(n)) \) truly random bits and has maximal entropy loss \( \Delta(k_{\text{max}}) = k_{\text{max}}/2 \).

### 4.1.4 Expander Graph Construction

Section 3.6 introduced an application of expander graphs and showed how to build input-restricting functions out of an expander. Furthermore, we showed in Theorem 3.27 that every conductor is actually an expander. Hence the Trevisan extracting conductor can be used to construct an expander graph. If we apply Theorem 3.27, we get the following expander graph:
Lemma 4.8. Let \( \text{Ext} \) be the strong extracting \((n, 0, k_{\max}) \times (d) \rightarrow 2_{\epsilon}(m, k_{\max}(k'))\) conductor from Section 4.1.3. Then \( \text{Ext} \) is also an \((2^{n}, 0, k_{\max} = 2^{k_{\max}}) \times (D) \rightarrow (2^{m} + d, \gamma = (1 - 2\epsilon)2^{d+m-k_{\max}})\) expander graph with left-degree \(D = 2^{d}\) where \(d = \psi(k_{\max}, \epsilon) \cdot l^{2} + l\).

The problem of this expander graph is that it has \(\text{superpoly}(n)\) degree \(D\) because of \(\ell^{2} = O(\log^{2}(n))\) and therefore, not interesting for the application in [MT07] presented in Section 1.2. We are going to introduce in Section 5.4 a construction of an expander graph which has only \(\text{poly}(n)\) degree and hence, is applicable for [MT07].

4.2 Constructing Conductors from Hash Functions

In this section, we investigate in extracting conductors which use a family of hash functions \(H = \{h : \{0, 1\}^{n} \rightarrow \{0, 1\}^{m}\}\) as seed instead of truly random bits. We can define a strong extracting conductor \(\text{Ext}_{SZ} : \{0, 1\}^{n} \times [H] \rightarrow \{0, 1\}^{m}\) where \(\text{Ext}_{SZ}(x, h) = h(x)\) and \(h\) randomly chosen from a class of hash functions \(H\).

In the following, we show how to get such a strong extracting conductor based on [SZ98] and additionally to [SZ98], we calculate the exact values of the seed length.

Let now \(H\) being a universal family of hash functions. A universal family \(H\) of hash functions is defined as a set of hash functions \(h : \{0, 1\}^{n} \rightarrow \{0, 1\}^{m}\) such that for two different \(x, y \in \{0, 1\}^{n}\), we have

\[
\frac{|\{h \in H : h(x) = h(y)\}|}{|H|} \leq 2^{-m}.
\]

We have the following well known lemma for universal families of hash function.

Lemma 4.9 (Leftover Hash Lemma [IZ89]). Let \(X\) be a \(k\)-source over \(\{0, 1\}^{n}\). Let \(H\) be a universal family of hash functions mapping \(\{0, 1\}^{n} \rightarrow \{0, 1\}^{m = k - \Delta}\). Then the distribution of \((h, h(x))\) is \(2^{-\Delta/2} = \epsilon/2\)-close to \(U_{u+m}\) if \((h, x)\) is chosen uniformly at random from \(H \times X\), where \(u\) are the number of bits needed to describe the family \(H\).

We could apply the Leftover Hash Lemma to show that \(\text{Ext}_{SZ}\) is indeed a strong extracting conductor, but the problem is that more than \(n\) bits are needed to describe such a universal family \(H\) of hash functions. Therefore, to avoid a big value of \(u\), we let \(H\) be a so called \(t\)-wise \(\rho\)-biased sample space and show that the statement of the Leftover Hash Lemma in [IZ89] is also valid for this \(t\)-wise \(\rho\)-biased sample space \(H\).

Definition 4.10 (\(t\)-wise \(\rho\)-biased sample space). A set \(S \subseteq \{0, 1\}^{\ell}\) is called a \(t\)-wise \(\rho\)-biased sample space \(S\) of \(\ell\)-bit vectors if for \(s \in \{0, 1\}^{\ell}\) chosen uniformly random from \(S\), we have that for all \(I \subseteq \ell\), with \(|I| \leq t\) and for all \(|I|-\text{bit strings} b\),

\[
|P_{s \in S}[s = b] - 2^{-|I|}| \leq \rho. \tag{4.2.1}
\]

We see the functions \(h \in H\) as \(\ell\) bit vectors of the sample space \(H\) with \(\ell = m2^{n}\), because every function \(f : \{0, 1\}^{n} \rightarrow \{0, 1\}^{m}\) can be described by \(m2^{n}\) bits. Further, we set \(\epsilon = 2^{1 - \Delta/2}\) where \(\Delta\) is a constant. As in the proof of Theorem 4.5, we fix the min-entropy to \(k_{\max}\) and assume that the output length \(m(k')\) is a function depending on the input min-entropy and describes the first \(m(k')\) output bits which are almost randomly distributed. Let \(m(k') := k_{m}(k') := k' - \Delta\) for \(k' \in [0, k_{\max}]\). Because we have the min-entropy fixed to \(k' = k_{\max}\), we
4.2. CONSTRUCTING CONDUCTORS FROM HASH FUNCTIONS

have the output length \( m = m(k_{\text{max}}) = k_{\text{max}} - \Delta \). We will argue later why we can generalize the results for min-entropies in \([0, k_{\text{max}}]\).

Let the parameters of the sample space \( H \) be \( t = 2m \) and \( \rho = 2^{-2m-\Delta}(\epsilon^2 2^\Delta - 1) \). Note that with this definitions, we have \( \rho \geq 0 \).

According to Lemma 2.12, it is sufficient to show that the collision probability of the distribution over \((h, h(x))\) is smaller than \( \frac{1 + \epsilon^2}{|H| \cdot 2^m} \) to prove that the output of \( \text{Ext}_{SZ} \) is \( \epsilon \)-close to \( U_d \circ U_m \).

Let \( A \) be the distribution of \((h, h(x))\). The collision probability of \( A \) is

\[
\text{col}(A) = \mathbb{P}_{x_1, x_2 \in X, h_1, h_2 \in H}[h_1 = h_2 \land h_1(x_1) = h_2(x_2)].
\]

Therefore,

\[
\text{col}(A) = \frac{1}{|H|} \mathbb{P}_{x_1, x_2 \in X, h \in H}[h(x_1) = h(x_2)]
\]

\[
\leq \frac{1}{|H|} \left[ \mathbb{P}_{x_1, x_2 \in X, h \in H}[x_1 = x_2] + \mathbb{P}_{x_1, x_2 \in X, h \in H}[h(x_1) = h(x_2) | x_1 \neq x_2]\right]
\]

\[
= \frac{1}{|H|} \cdot 2^{k_{\text{max}}} + \frac{1}{|H|} \mathbb{P}_{x_1, x_2 \in X, h \in H}[h(x_1) = h(x_2) | x_1 \neq x_2]
\]

\[
\leq \frac{1}{|H|} \cdot 2^{k_{\text{max}}} + \max_{x_1 \neq x_2} \mathbb{P}_{h \in H}[h(x_1) = h(x_2)]. \tag{1}
\]

For any \( x_1 \neq x_2 \in X \) we have

\[
\frac{1}{|H|} \mathbb{P}_{h \in H}[h(x_1) = h(x_2)] = \frac{1}{|H|} \sum_{b \in \{0,1\}^m} \mathbb{P}_{h \in H}[h(x_1) = h(x_2) = b]
\]

\[
\leq \frac{1}{|H|} \sum_{b \in \{0,1\}^m} (2^{-t} + \rho)
\]

\[
= \frac{1}{|H|} \sum_{b \in \{0,1\}^m} (2^{-2m} + 2^{-2m-\Delta}(\epsilon^2 2^\Delta - 1))
\]

\[
= \frac{1}{|H|} \cdot 2^m \cdot (2^{-2m} + 2^{-2m}(\epsilon^2 - 2^\Delta))
\]

\[
= \frac{1}{|H|} \cdot 2^m (1 + \epsilon^2 - 2^{m-k_{\text{max}}})
\]

\[
= \frac{(1 + \epsilon^2)}{|H| \cdot 2^m} - \frac{1}{|H| \cdot 2^{k_{\text{max}}}).
\]

Inserting in Equation (1) leads to \( \text{col}(A) \leq \frac{(1 + \epsilon^2)}{|H| \cdot 2^m} \leq \frac{(1 + \epsilon^2^2)}{|H| \cdot 2^m}. \) Thus, the distribution \( A \) over \((h, h(x))\) is \( \epsilon \)-close to \( U_d \circ U_m \).

At this point, we showed that \( \text{Ext}_{SZ} \) is a strong extracting \((n, k_{\text{max}}, k_{\text{max}})\times (u) \rightarrow (m, k' \rightarrow k' - \Delta)\) conductor for a constant \( \Delta \). To see that \( \text{Ext}_{SZ} \) is also a strong extracting \((n, 0, k_{\text{max}})\times (u) \rightarrow (m, k_m(k') = k' - \Delta)\) conductor for \( k' \in [0, k_{\text{max}}] \), we show that the min-entropy is condensed in the first \( k_m(k') \) output bits.

Recall that we assumed \( m \) being a function depending on \( k' \) and have set \( m(k') = k_m(k') \).

Let now \( k' = k_{\text{max}} - r \) for a \( r \). Thus, we see \( m(k') = k' - \Delta \) as a function depending on \( r \) and have \( m(r) = k_{\text{max}} - r - \Delta = m - r \).
Then, we ignore the last $r$ bits from the output $h(x)$. We will show now that the new sample space $H'$ containing description strings of functions $h' : \{0, 1\}^n \to \{0, 1\}^{m-r}$ still fulfills the requirement of being a $(r)$-wise $\rho(r)$-biased sample space with $t(r) = 2(m - r)$ and $\rho(r) = 2^{-2(m-r) - \Delta} (e^{2\Delta} - 1)$. Note that $t = t(0)$ and $\rho = \rho(0)$.

We have to show that the difference between the collision probability of $H'$ and the collision probability of $U_{m-r}$ is less than $\rho(r)$. To achieve this, we will show a stronger result which implies a small collision probability.

Let $b = h'_2(x_2) = y_2$ for a $h'_2 \in H'$. We get for fixed $x_1, x_2 \in X$ and $y_1, y_2 \in \{0, 1\}^{m-r}$

$$
\left| P_{h'_1 \in H} \left[ h'_1(x_1) = y_1 \land h'_2(x_2) = y_2 \land y_1 = y_2 \right] - 2^{-2(m-r)} \right|
\leq \sum_{y_1, y_2 \in \{0, 1\}^r} \rho(0)
\leq 2^{2r} \cdot \rho(0)
\leq 2^{-2(m-r) - \Delta} (e^{2\Delta} - 1) = \rho(r)
$$

where at Step (2) we used the fact that $H$ is a $t(0)$-wise $\rho(0)$-biased sample space. Hence, $H'$ fulfills the requirement of being a $(r)$-wise $\rho(r)$-biased sample space and thus, for every $h' \in H'$ and $x \in X$, $h'(x)$ is $\epsilon$-close to $U_{m-r}$.

Therefore, the output $\text{Ext}_{SZ}$ contains still $(m-r)$ min-entropy which is the value of $k_m(k')$ for $k' = k_{\text{max}} - r$ and $\text{Ext}_{SZ}$ is a $(n, 0, k_{\text{max}}) \times (u) \to (m, k' \mapsto k' - \Delta)$ conductor.

Finally, we will show that the number of bits needed to describe the sample space $H$ is smaller than $n$. To get this number of bits $u$, we apply a result of [AGHP90] which gives an upper bound of the needed bits to describe the $t$-wise $\rho$-biased sample space $H$.

**Lemma 4.11.** Let $t$ be an odd integer. A $t$-wise $\rho$-biased sample space $H$ of $t$-bit vectors can be fully described by using

$$
2 \cdot \left[ \log(1/\rho) + \log \left(1 + \frac{t-1}{2} \cdot \log(\ell + 1)\right) \right]
$$

bits, such that there exists an efficient algorithm calculating the $h \in H$ for a specific $x \in X$.

Inserting our parameters gives

$$
\begin{align*}
\log(1/\rho) + \log \left(1 + \frac{t-1}{2} \cdot \log(\ell + 1)\right) & = 2 \cdot \left[ \log \left(2^m \cdot (e^2 - 2^{-\Delta})^{-1} \right) + \log \left(1 + \frac{2m-1}{2} \cdot \log(m2^n + 1)\right) \right] \\
& = 2 \cdot \left[ 2m - \log \left(e^2 - 2^{-\Delta}\right) + \log \left(1 + \frac{2m-1}{2} \cdot \log(m2^n + 1)\right) \right] \\
& = 2 \cdot \left[ 2m + \Delta - \log 3 + \log \left(1 + \frac{2m-1}{2} \cdot \log(m2^n + 1)\right) \right] \\
& = 2 \cdot \left[ \Delta + \log \left(1 + \frac{2m-1}{2} \cdot \log(m2^n + 1)\right) \right] + 4m - \log 9 \\
& \approx [2\Delta] + 4 \log m + 2 \log n + 4m - \log 9 \\
& \leq 6 \log n + 4m + 4\Delta = 6 \log n + 4k_{\text{max}}
\end{align*}
$$
4.3. STRONG CONDENSING CONDUCTOR

Overall, we get the following strong extracting conductor $\text{Ext}_{SZ}$:

**Theorem 4.12.** For every $n, k_{\text{max}} \leq n$ let $\Delta = 2 \cdot \log(1/\epsilon) + 2$ be the constant entropy loss and therefore $\epsilon = 2^{1-\Delta/2}$. Then there exists an explicit strong extracting $(n, 0, k_{\text{max}}) \times (d) \rightarrow \epsilon$ $(m, k' \mapsto k' - \Delta)$ conductor $\text{Ext}_{SZ} : \{0, 1\}^n \times [H] \rightarrow \{0, 1\}^m$ for $k' \in [0, k_{\text{max}}]$ and with $m = k_{\text{max}} - \Delta$ and seed length $d = 6 \log n + 4k_{\text{max}}$.

**Note.** Because we need $O(k + \log n)$ truly random bits, using this conductor makes sense only if the source min-entropy $k$ is relatively small, e.g. $k \in O(\log n)$.

4.3 Building a Strong Condensing Conductor from Trevisan’s Conductor

Before we analyze the lossless conductor introduced in [TUZ01], we discuss a special property of the Trevisan extracting conductor $\text{Ext}_T$ and show afterward how to get the wished strong condensing conductor.

4.3.1 Reconstructive Extracting Conductors

In Section 4.1, we showed that the Trevisan’s construction leads to an injective strong extracting conductor $\text{Ext}_T$ with the following method: If a distinguisher $D_{\epsilon}$ exists which can distinguish the output of $\text{NW}_{x, \bar{x}}(U_d)$ from $U_m$ with success at least $\epsilon$ than we can compute a short advice string and with the help of $D_{\epsilon}$ and this advice string, we can compute a function $g$ which approximates function $\bar{x}$ over a fraction of the domain of at least $1/2 + \epsilon/m$. I.e. function $g$ is an EC encoding with less than $1/2 + \epsilon/m$ relative hamming distance to the encoding $\bar{x}$. We than argued that the number of possible $g$’s and the number of $x$ with $\bar{x}$ having hamming distance smaller than $1/2 + \epsilon/m$ to $g$ are relatively small and if we choose the parameters right, we get an extracting conductor. But this implies that if we set the parameters badly than the probability to choose a bad $x$ will be high. Choosing a bad $x$ implies that there exist a short advice string describing an approximation of $\bar{x}$, given a distinguisher $D_{\epsilon}$ and because the EC encoding can be efficiently decoded, we could actually reconstruct $x$ with the help of the distinguisher $D_{\epsilon}$ and the short advice string. The reconstruction would be achieved by converting the distinguisher to a next-bit predictor and then guessing $x$ bit by bit. Hence, $\text{NW}_{x, \bar{x}}(U_d)$ cannot be a good pseudo-random generator and thus, $\text{Ext}_T$ cannot be an extracting conductor.

We present now an abstraction of conductors which prove there extracting property through the above argument.

**Definition 4.13** (reconstructive extracting conductor). Let

- $E : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$
- $A : \{0, 1\}^n \times \{0, 1\}^{d_A} \rightarrow \{0, 1\}^a$
- $R^T : \{0, 1\}^a \times \{0, 1\}^{d_A} \times \{0, 1\}^r \rightarrow \{0, 1\}^n$

6Recall that every strong condensing conductor is a lossless conductor
7Recall that $\bar{x}$ is an EC encoding of input $x$ interpreted as a truth table of a boolean function $\bar{x}$
CHAPTER 4. BASIC CONSTRUCTIONS

be functions, called the extractor, advice and reconstruction functions, respectively. Then we call the triple \((E, A, R)\) a \((p, q)\)-reconstructive extracting conductor if for every distribution \(X\) over \(\{0,1\}^n\) and every next-bit predictor \(T : \{0,1\}^m \rightarrow \{0,1\}\) for \(E(X, U_d)\) with success \(p\), we have

\[
P_{x \in X, y \in U_d, z \in U_r}[R^T(A(x, y), y, z) = x] \geq q,
\]

where \(R^T\) means that the function \(R\) makes black-box calls to the next-bit predictor \(T\) for reconstructing input \(x\).

Function \(A\) calculates the advice string of length \(a\) for every \(x \in \{0,1\}^n\) and function \(R^T\) reconstructs \(x\) with the help of the advice string if there exists a next-bit predictor \(T\) with success at least \(p\). Generally, the function \(R^T\) takes additional randomness of length \(r\) to reconstruct the value of \(x\). Note that we use a next-bit predictor rather than a distinguisher for this definition. According to Lemma 2.15, we know that every distinguisher can be transformed to a next-bit predictor but with a loss in the advantage. Therefore, we have defined reconstructive extracting conductors directly with a next-bit predictor to avoid this loss.

The main idea of the conductor construction of this section is the observation that the advice function of a reconstructive extracting conductor is actually a strong condensing conductor if the extractor function \(E\) is not an extracting conductor. This can be seen by the following consideration: If \(E\) is not an extracting conductor, then \(R^T\) can reconstruct \(x\) (which was chosen from a \(k\)-source) with the help of the short \(a\)-bit advice string calculated by \(A\). Therefore, the advice string must still contain \(k\) min-entropy and with \(a < n\) we non-trivially condensed the randomness of \(x\). We will show how to choose the parameters of the extractor function \(E\) such that it is not an extracting conductor and such that \(R^T\) can fully reconstruct the original input \(x\).

This is formalized by the following lemma.

**Lemma 4.14.** Let \((E, A, R)\) be a \((p, q = 1 - \epsilon)\) reconstructive extracting conductor and \(X \subset \{0,1\}^n\) a subset with \(|X| = k\) such that there exists a next-bit predictor \(T : \{0,1\}^m \rightarrow \{0,1\}\) for \(E(X, U_d)\) with success \(p\). Then the distribution \(U_{d_A} \circ A(X, U_{d_A})\) is \(2\epsilon\)-close to a distribution \(U_{d_A} \circ D\) with \(d_A + k\) min-entropy, that is, \(A\) is a strong condensing \((n, k, k) \times (d_A) \rightarrow (m, k' \rightarrow k')\) conductor.

**Proof.** Let \(G\) be the set of good pairs such that \((x, y) \in G\) if

\[
P_z[R^T(A(x, y), y, z) = x] > 1/2
\]

The Equation (4.3.1) implies \(A(x_1, y) \neq A(x_2, y)\) if both pairs \((x_1, y)\) and \((x_2, y)\) are in \(G\). In particular, we get a bijective mapping \(A'\) on the set \(G\) if we define \(A'(x, y) = A(x, y) \circ y\).

Furthermore, we have

\[
P_{x \in X, y \in G}[R^T(A(x, y), y, z) = x] \geq 1 - \epsilon.
\]

We show now by contradiction that \(P_{x \in X, y \in G}(x, y) \in G\) \(\geq 1 - 2\epsilon\) must hold. We know

\[
P_{x \in X, y}[R^T(A(x, y), y) = x] = P_{x \in X, y}[x, y) \in G] \cdot P_{x \in X, y}[R^T(A(x, y), y) = x| (x, y) \in G] \\
+ P_{x \in X, y}[x, y) \notin G] \cdot P_{x \in X, y}[R^T(A(x, y), y) = x| (x, y) \notin G] \\
\leq P_{x \in X, y}[x, y) \in G] \\
+ P_{x \in X, y}[x, y) \notin G] \cdot P_{x \in X, y}[R^T(A(x, y), y) = x| (x, y) \notin G],
\]

54
4.3. STRONG CONDENSING CONDUCTOR

We assume that \( P_{x \in X, y \in G}[(x, y) \in G] \) is strict smaller than \( 1 - 2\epsilon \) and therefore, \( P_{x \in X, y \in G}[(x, y) \notin G] \geq 2\epsilon \) must hold. Furthermore, we can conclude from the definition of being a good pair that \( P_{x \in X, y \in G}[R^T(A(x, y), y) = x |(x, y) \notin G] \leq 1/2 \). This gives

\[
P_{x \in X, y}[R^T(A(x, y), y) = x] < (1 - 2\epsilon) + 2\epsilon \cdot \frac{1}{2} = 1 - \epsilon
\]

which contradicts Equation (4.3.2) and therefore \( P_{x \in X, y \in G}[(x, y) \in G] \geq 1 - 2\epsilon \) must hold. Hence, almost all but a fraction of \( 2\epsilon \) of the pairs on \( X \times U_{d_A} \) are in \( G \) and therefore \( 2\epsilon \)-close to the distribution \( X \times U_{d_A} \). Because \( A'(X, U_{d_A}) \) is a bijective mapping onto \( G \), we can conclude that \( A'(X, U_{d_A}) \) is also \( 2\epsilon \)-close to \( X \circ U_{d_A} \) and has min-entropy \( k + d_A \). \( \square \)

In the next section, we present a concrete strong condensing conductor constructed out of Trevisan’s conductor.

4.3.2 Strong Condensing Conductor Construction

We introduce now a strong condensing conductor which will be used as a basic building block for the extracting conductor of Section 5.3. First, we state that the Trevisan conductor is indeed a reconstructive extracting conductor and second, we give an advice function \( A \) such that it is a strong condensing conductor.

**Lemma 4.15.** Let \( \text{Ext}_{\mathcal{F}, EC} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m \) be the function of Section 4.1 with \( m = \frac{k_{\text{max}} + d}{\epsilon} \). Then there exist a function \( A \) and a function \( R \) such that \( (\text{Ext}_{\mathcal{F}, EC}, A, R) \) is a \( (1 - \epsilon, 1 - 10\epsilon) \) reconstructive extracting conductor.

The proof will be given at the end of this section.

We will develop now the final strong condensing conductor step by step: First, we construct a strong condensing conductor \( A : \{0,1\}^n \times \{0,1\}^{d_A} \rightarrow \{0,1\}^n \), and second, we take this conductor \( A \) and iteratively cascade it with itself to reduce the final output length.

The basic conductor \( A \) is constructed with the help of Trevisan’s conductor, or more precisely, conductor \( A \) will be the advice function of the reconstructive extracting conductor of Lemma 4.15. We fix an EC code \( EC : \{0,1\}^n \rightarrow \{0,1\}^\rho \) with relative distance \( > 0.1 \) and we fix a weak \( (m, d, \ell, \rho) \) design \( \mathcal{F} = \mathcal{F}_1, \ldots, \mathcal{F}_m \) for \( m = \frac{k_{\text{max}} + d}{\epsilon} \). We will give the value of \( \rho \) later when we compute the value of \( d \).

The advice function \( A \) makes the following calculation steps described in Algorithm 4.3.1.

We see that the output of \( A \) has maximal length of \( \sum_{j \leq 1} 2^{|S_j \cap S_j|} \leq pm \) because of the used weak design and \( d_A = \log m + d - \ell \). Because \( \log m < \ell \) we have \( d_A \leq d \) and can therefore use the same randomness space \( \{0,1\}^d \) for the advice function \( A \) as for the extracting function \( E \).

The reconstruction function \( R^T \) would then take the output \( out \) of function \( A \) to reconstruct the input \( x \). To achieve this, function \( R^T \) reconstructs the indices \( j \) and \( \gamma \) used in Algorithm 4.3.1 to find the corresponding evaluations of function \( \bar{x}(\cdot) \) in the output \( out \) of function \( A \). With the help of the next-bit predictor \( T \), function \( R \) can reconstruct \( \bar{x} = EC(x) \) and hence \( x \). The detailed description on how the reconstruction function \( R^T \) works, can be found in the proof of Lemma 4.15 at the end of this section.

The next lemma states that the advice function \( A \) described in Algorithm 4.3.1 is indeed an injective strong condensing conductor.
CHAPTER 4. BASIC CONSTRUCTIONS

Algorithm 4.3.1 Advice function \( A(x,u) \)

\[
\text{out} \quad \text{// output of } A
\]

\[
\text{calculate } \bar{x} = EC(x)
\]

Interpret \( u \) as follows: first \( \log m \) bits describe \( i \in [m] \) and the next \( (d - \ell) \) bits describe a \( \beta \in \{0,1\}^{d-\ell} \).

\textbf{for all } \ j < i \text{ and } \gamma \in \{0,1\}^{\mid S_i \cap S_j \mid} \text{ do}

\[
\text{// construct a new string } y \in \{0,1\}^d \text{ as follows:}
\]

\[
\text{set } \beta \text{ at positions } [d] \setminus S_i \text{ of } y \\
\text{set } \gamma \text{ at positions } S_i \cap S_j \text{ of } y \\
\text{fill remaining positions of } y \text{ with zeros}
\]

\[
\omega(j,\gamma) := y\mid_{S_j}
\]

out := out || \( \bar{x}(\omega(j,\gamma)) \)

\textbf{end for}

\textbf{return} out

Lemma 4.16. Let \( \mathcal{F} \) be a weak \((m,d,\ell,\rho)\) design with \( m \geq \frac{k_{\max} + d}{\epsilon} \) and \( \ell = \log(n) \). Then function

\[
A : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^{m_a}
\]

as described above is an explicit and injective strong condensing \((n,0,k_{\max}) \times (d) \rightarrow 20\epsilon \times (m_a,k' \mapsto k')\) conductor with \( m_a = \rho m = \rho \frac{k_{\max} + d}{\epsilon} \).

\textbf{Proof.} We choose the reconstructive extracting conductor from Lemma 4.15 and apply Lemma 4.14. Hence, \( A \) is a strong condensing \((n,0,k_{\max}) \times (d) \rightarrow 20\epsilon \times (m_a,k' \mapsto k')\) conductor. Because Trevisan’s conductor is explicit and Algorithm 4.3.1 can be efficiently computed, we can conclude that \( A \) is an explicit conductor.

That \( A \) is actually a strong condensing \((n,0,k_{\max}) \times (d) \rightarrow (m_a,k' \mapsto k')\) conductor can be seen by analyzing the calculation steps which \( A \) does: As in Trevisan’s construction of a conductor, conductor \( A \) uses weak designs to calculate the projections \( y|_{S_j} \) and as we already argued in the proof of Theorem 4.5, the conductor construction works for all input min-entropies \( k' \in [0,k_{\max}] \). Thus, \( A \) is a strong condensing \((n,0,k_{\max}) \times (d) \rightarrow (m_a,k' \mapsto k')\) conductor.

It remains to show that \( A \) is injective. We show it similarly as for the Trevisan conductor. Assume that \( A \) is not injective. This would lead to

\[
\exists x' \neq x \in \{0,1\}^n : \forall u \in \{0,1\}^d : A(x,u) = A(x',u).
\]

Function \( A \) constructs for each input \( u \) 2\( d \) strings \( y \) and calculates \( \bar{x}(y|_{S_j}) \) for all \( j < i \). We fix a \( j \), then for being non-injective, we have

\[
\bar{x}(\omega(j,\gamma)) = \bar{x}'(\omega(j,\gamma)) \quad \forall \gamma \in \{0,1\}^{\mid S_i \cap S_j \mid}
\]
4.3. STRONG CONDENSING CONDUCTOR

and in particular
\[ x(y_{S_j}) = x'(y_{S_j}). \]

We see that by iterating through all \( u \in \{0, 1\}^d \) and \( \gamma \in \{0, 1\}|S \cap S'| \), we called the functions \( \bar{x}(\cdot) \) and \( \bar{x}'(\cdot) \) with all possible inputs in \( \{0, 1\}^\ell \) and hence non-injectivity of \( A \) would imply equality of the truth tables of \( \bar{x}(\cdot) = EC(x) \) and \( \bar{x}'(\cdot) = EC(x') \) which contradicts the property of the EC encoding because \( x \neq x' \). Hence, such an \( x' \) cannot exist and conductor \( A \) must be an injective function.

The value of \( d \). We choose \( 1 < \rho = e^{\alpha \ell} \) for a constant \( \alpha > 0 \). From Lemma 4.7 we know that there exists an efficiently computable weak design with \( d = \left\lceil \frac{\ell}{n \rho} \right\rceil \cdot \ell = \left\lceil \frac{\ell}{\alpha} \right\rceil \). Recall that the length of the EC encoding is \( 2^\ell \). We will assume that we have \( \ell = s \cdot \log n \) for a constant \( s \) and \( n \) being sufficiently big. We get

\[ m_a = \rho m = e^{\alpha \ell} \cdot \frac{k_{\text{max}} + \ell/\alpha}{\epsilon} = n^{s \cdot \alpha \log e} \left( \frac{k_{\text{max}} + s \cdot \log n/\alpha}{\epsilon} \right). \]

To reduce the output length \( m_a \) to a length \( m_a = O \left( (k_{\text{max}})^{1+2\delta} \right) \) for a constant \( \delta \), we will cascade the strong condensing conductor \( A \) with itself several times as described in the end of this section. This leads to the following final strong condensing conductor \( C_{TUZ} \):

**Theorem 4.17** (final strong condensing conductor). For every \( n \), \( k_{\text{max}} \leq n \), a constant \( \epsilon \in (0, 1) \) and a constant \( \delta > 0 \), there exists an explicit strong injective condensing \( (n, 0, k_{\text{max}}) \times (d) \to (n', k' \mapsto k') \) conductor \( C_{TUZ} \) with \( d = (2 + \delta) \cdot \left\lceil \frac{\ell}{\alpha} \right\rceil \) and \( n' = \left( \frac{k_{\text{max}} + \ell/\alpha}{\epsilon} \right)^{1+2\delta} \), where \( \ell = s \cdot \log n \) for a constant \( s \) and \( \alpha = \frac{\delta}{2+\delta} \cdot \frac{1}{s \log e} \).

The proof is given at the end of this section.

**Proof of Lemma 4.15.** We will present an adapted version of the proof given in [TUZ01]. We first prove that a next-bit predictor \( T \) exists if \( m \geq k_{\text{max}} + d \).

To show that such a next-bit predictor \( T \) exists for the Trevisan extracting conductor \( Ext_T(X, U_d) \), we use Lemma 2.16 in the following way: Let \( Y \) be the distribution of \( Ext_T(X, U_d) \). We know that \( H_\infty(Ext_T(X, U_d)) \leq H_\infty(X) + H_\infty(U_d) \leq k_{\text{max}} + d \) must hold. Furthermore, to get the existence of a next-bit predictor \( T \) with success \( 1 - \epsilon \), we have to fulfill

\[ \frac{1}{m} H_\infty(Ext(X, U_d)) \leq \frac{k_{\text{max}} + d}{m} \leq \epsilon \]

which is done by setting \( m \geq \frac{k_{\text{max}} + d}{\epsilon} \). Therefore, a next-bit predictor \( T \) with success \( 1 - \epsilon \) exists for \( Ext_T \) being the extracting function.

It remains to show that

\[ P_{x \in x \in U_d} [R_T(A(x, y), y) = x] \geq 1 - 10\epsilon \]

holds.

We choose our advice function \( A \) to be the function described in Algorithm 4.3.1. Recall that we have \( \bar{n} = 2^\ell \). The reconstruction function we will describe, will not need additionally randomness and thus, we set \( r = 0 \). Let the reconstruction function \( \bar{R}_T : \{0, 1\}^n \times \{0, 1\}^d \to \{0, 1\}^\ell \)
\textbf{Algorithm 4.3.2} Reconstruction function $R^T(out, u)$

interpret $u = u_1||u_2||u_3$ with $|u_1| = \log m$ and $|u_2| = d - \ell$

set $i \in [m]$ // defined by $u_1$

set $\beta := u_2$

\textbf{for all} $a \in \{0,1\}^\ell$ \textbf{do}

// construct a new string $y \in \{0,1\}^d$ as follows:

set $\beta$ at positions $[d]\setminus S_i$ of $y$

set $a$ at positions $S_i$ of $y$

set $\gamma_j := y_{S_i \cap S_j}$ for all $1 \leq j < i$

set $w_a = T_{X,y} (out_1,\cdots,out_i-1,\gamma_i-1) = T_{X,y} (\bar{x}(y|S_1),\cdots,\bar{x}(y|S_{i-1}))$

\textbf{end for}

$w := w_1w_2\cdots w_2^\ell$ // view $w$ as a bit string in $\{0,1\}^n$

\textbf{if} $\exists$ unique $x \in \{0,1\}^n$ s.t. $\text{dist}(\bar{x},w) > 0.1$ \textbf{then}

return $x$

\textbf{else}

return $\perp$

\textbf{end if}
\{0,1\}^n$ be a function which does the steps defined in Algorithm 4.3.2, where $dist(\cdot, \cdot)$ is the normalized Hamming distance. Note that the output $out$ of $A$ is indexed by $1 \leq j < i$ and a $\gamma \in \{0,1\}^{S_i \cap S_j}$.

From Lemma 2.16 we know that

$$P_{x \in X, y \in U_{d,i}}[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i}) \geq 1 - \epsilon.$$  \hspace{1cm} (4.3.3)

Note that the values of $\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}})$ are known because there are described in the advice string.

Let $G$ be the set of good pairs $(X, U_d)$. We say that $(x, y) \in G$ if

$$P_i[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i}) \geq 0.9 \hspace{1cm} (4.3.4)$$

We show by contradiction that $P_{x \in X, y \in U_{d,i}}[(x, y) \in G] \geq 1 - 10\epsilon$ must hold. We have

$$P_{x,y,i}[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i})] =$$

$$P_{x,y,i}[(x, y) \in G] \cdot P_{x,y,i}[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i})(x, y) \in G]$$

$$+ P_{x,y,i}[(x, y) \notin G] \cdot P_{x,y,i}[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i})(x, y) \notin G]$$

$$\leq P_{x,y,i}[(x, y) \in G] + P_{x,y,i}[(x, y) \notin G]$$

$$\cdot P_{x,y,i}[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i})](x, y) \notin G]$$

To contradict, we assume that $P_{x,y,i}[(x, y) \in G] < 1 - 10\epsilon$ and thus, $P_{x,y,i}[(x, y) \notin G] \geq 10\epsilon$ must hold. Additionally, we can conclude from the definition of being a good pair that $P_{x,y,i}[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i}) \notin G] \leq 1/10$. This leads to

$$P_{x,y,i}[T_{X,y}(\bar{x}(y_{S_1}), \cdots, \bar{x}(y_{S_{n-1}}))] = \bar{x}(y_{S_i})] < (1 - 10\epsilon) + 10\epsilon \cdot \frac{1}{10} = 1 - 9\epsilon$$

which contradicts Equation (4.3.4) and therefore $P_{x \in X, y \in U_{d,i}}[(x, y) \in G] \geq 1 - 10\epsilon$ must hold.

For every $a \in \{0,1\}^k$ we have $w_{a} \equiv \bar{x}(a) = \bar{x}(y_{S_i})$, where $\equiv$ should denote that it is a guessing, and hence, after doing guesses for all possible $a$, we have $w \equiv \bar{x}$. If we used for the Trevisan conductor an EC encoding with relative distance > $0.1$ we can conclude that $R^T$ outputs $x$ with probability at least $1-10\epsilon$ because for $(x, y) \in G$ we know that $dist(\bar{x}, w) \leq 0.1$ and that $\bar{x}$ represent an encoding for a unique $x$ in this Hamming ball. Hence, we have

$$P_{x,y,i}[R^T(A(x, y), y) = x] \geq 1 - 10\epsilon$$

**Proof of Theorem 4.17.** We show how to get the final strong condensing conductor of Theorem 4.17 by composing the strong condensing conductor $A$ from Lemma 4.16 with itself several times. The problem of conductor $A$ is the too long output length $m_a$. Because $m_a$ contains still $k$ min-entropy we could just apply conductor $A$ again on the output $m_a$ and get a new output length $< m_a$. This reduction is not for free, indeed we add an additional error of $\epsilon$ with each iteration step. In the following Lemma 4.18 we present such a construction.

**Lemma 4.18 (iterated condensing conductor cascading).** If $C = \{C_n : \{0,1\}^n \times \{0,1\}^{d(n)} \rightarrow \{0,1\}^{m(n)}\}$ is a family of (strong) condensing $(n, k_{\text{min}}, k_{\text{max}}) \times (d(n)) \rightarrow \epsilon (m(n), k' \mapsto k')$ conductors, they can be composed repeatedly. Let $n_1$, $k_{\text{min}}$, $k_{\text{max}}$ and $\epsilon > 0$ be given, then we define $C^{(1)} := C_{n_1}$ and for $i > 1$

$$C^{(i)} := C_{n'} \circ C^{(i-1)},$$

59
where \( n' \) is the output length of \( C^{(i-1)} \). If \( m(n) \leq n^a \omega \) for a fixed \( a < 1 \) and \( \omega > 0 \), \( d(n) = b \cdot \log n \leq d(n_1) \) for a constant \( b \) and \( n' \leq n_1 \) for all \( n' \), then for all \( i \geq 1 \), \( C^{(i)} \) is a (strong) condensing \((n_1, k_{min}, k_{max}) \times (d) \rightarrow_{\epsilon} (m, k' \mapsto k')\) conductor with

- \( m \leq \omega^{\frac{1}{1-a}} \cdot n_1^{a^i} \)
- \( d \leq \frac{ib}{1-a} \cdot \log(\omega) + \frac{b}{1-a} \cdot \log n_1 \)

**Proof.** A special version of the above iterated cascading was presented in [TUZ01] for the case where \( k_{min} = k_{max} \) and we use the same method to prove our generalized iterated cascading lemma.

We compose the conductors according to the cascading defined in Lemma 3.19, hence, the final error is \( i \cdot \epsilon \). Let \( d_i \) be the seed length and \( n_i \) be the input length of \( C^{(i)} \). Further, we have for \( i > 1 \) that \( n_i \leq \omega \cdot n_{i-1}^{a} \) with \( a < 1 \). Thus

\[
m \leq n_i \leq \omega \omega^a \cdot \omega^a \cdot \cdots \omega^a \cdot n_1^{a_i} \leq \omega^\frac{1}{1-a} \cdot n_1^{a_i},
\]

where we used \( \sum_{j=0}^{\infty} q^j = \frac{1}{1-q} \) for a \( q < 1 \).

For the seed length \( d \) we get

\[
d = \sum_{j=1}^{i} d(n_j) \leq b \cdot \sum_{j=1}^{i} \log(n_j) \leq b \cdot \sum_{j=1}^{i} \log \left( \omega^\frac{1}{1-n} \cdot n_1^{a^i} \right)
\]

\[
\leq ib \cdot \frac{1}{1-a} \cdot \log(\omega) + b \cdot \sum_{j=1}^{i} (a^j) \cdot \log n_1
\]

\[
\leq \frac{ib}{1-a} \cdot \log(\omega) + \frac{b}{1-a} \cdot \log n_1
\]

\( \square \)

To get the wished conductor of Theorem 4.17, we cascade \( i \) times conductor \( A \) of Lemma 4.16 with itself and set the error of conductor \( A \) to \( \epsilon_A = \frac{\epsilon}{i} \). Further, we have \( \ell = s \cdot \log n \) for a constant \( s \) and we set \( \alpha = \frac{\delta}{2+\delta} \cdot \frac{1}{s \cdot \log e} \) for a constant \( \delta \). Note that \( 3/2 \leq \rho = e^{\alpha \ell} \) is still fulfilled for \( n \) big enough. Furthermore, we set \( a = s \cdot \alpha \log e = \frac{\delta}{2+\delta} \). In particular, we have \( m_a = m(n) = n^a \cdot \alpha \log e \left( \frac{k_{max} + s \cdot \log n/\alpha}{\epsilon_A} \right) = n^a \cdot \omega \) for \( \omega := \left( \frac{k_{max} + s \cdot \log n/\alpha}{\epsilon_A} \right) \). Let \( n \) be the initial input length \( n_1 \) and \( \ell := \ell(n) \).

We choose \( i := \frac{2 \log n}{\delta \log \omega} \) and get therefore

\[
a_i = \frac{\delta \log \omega}{2 \log n}, \quad (1)
\]

Furthermore, we have

\[
\frac{1}{1-a} = \frac{1}{1-\frac{\delta}{2+\delta}} = 1 + \frac{\delta}{2} \quad (2)
\]

60
4.3. STRONG CONDENSING CONDUCTOR

We get for the final output length $n'$:

$$n' \leq \omega \frac{n^{1 - a}}{\alpha} \cdot n^a \quad (1)$$

$$= \omega \frac{n^{\frac{k_{\max}}{\epsilon}}}{\alpha} = \omega \frac{n^{\frac{i}{\alpha}} \log n}{\epsilon} \quad (2)$$

$$= \omega^{1 + \delta/2} \cdot \omega^{\delta/2} = \omega^{1 + \delta}.$$

Additionally, we have $i \ll \left(\frac{k_{\max} + \ell / \alpha}{\epsilon}\right)$. Therefore,

$$n' \leq \omega^{1 + \delta} = \left(\frac{k_{\max} + \ell / \alpha}{\epsilon A}\right)^{1 + \delta} = \left(\frac{i(k_{\max} + \ell / \alpha)}{\epsilon}\right)^{1 + \delta} \leq \left(\frac{k_{\max} + \ell / \alpha}{\epsilon}\right)^{1 + 2 \cdot \delta}.$$

For the value of $d$ we get

$$d \leq \frac{i s}{\alpha} \cdot \frac{1}{1 - a} \cdot \log(\omega) + \frac{s}{\alpha} \cdot \frac{1}{1 - a} \cdot \log n$$

$$\leq \frac{s \cdot \log n}{\alpha} \cdot \frac{1}{1 - a} + \frac{1}{1 - a} \cdot \frac{s \cdot \log n}{\alpha}$$

$$= \frac{2}{1 - a} \cdot \ell / \alpha \quad (3)$$

where we set $b = s / \alpha$ and at Step (3) we used $i = \log \frac{2 \log n}{\delta \log \omega} \leq \frac{2 \log n}{\delta \log \omega}$.

We got the wished conductor.
5 Compositions of Basic Constructions

In the previous chapters we have introduced basic conductor constructions as well as strong composition theorems. In this chapter, we will discuss possible combinations of the basic conductors. In Section 5.1, we show how to reduce the error of the Trevisan construction of Section 4.1 and in Section 5.2 we will get an almost ideal conductor with respect to the entropy loss with the help of the conductors of Sections 5.1 and 4.2. In Section 5.3, we present an improvement of this almost ideal conductor in the sense of using a shorter seed. Finally, in Section 5.4 we construct an expander graph fulfilling the requirements needed to be useful for the application described in Section 1.2 which is one of the main purposes of this thesis.

5.1 Iterated Concatenation of Trevisan's Conductor

In Section 4.1.3 we presented a concrete instantiation of Trevisan’s conductor construction with entropy loss $\Delta(k_{\text{max}}) = k_{\text{max}}/2$. We show now how to reduce the entropy loss by iteratively applying the concatenation composition of Section 3.3.2 to this conductor with itself. The goal is to reduce the entropy loss to $\Delta(k') = \log(k')$. The idea of iteratively applying Trevisan’s conductor to get a small entropy loss was mentioned in [MST04]. We give now a detailed construction based on this idea. We have the strong extracting $(n, 0, k_{\text{max}}) \times (d_1) \rightarrow \epsilon_1 (m_1, k_1(k'))$ conductor $\text{Ext}_1$ with $m_1 = k_{\text{max}}/2$, $k_1(k') = (k' - 3\log(m_1/\epsilon_1) - 1)/\rho_1$ for $k' \in [0, k_{\text{max}}]$ and entropy loss $\Delta(k_{\text{max}}) = k_{\text{max}}/2$. Clearly, Trevisan’s conductor can also be instantiated as a strong extracting $(n, 0, k_{\text{max}}/2-1) \times (d_1) \rightarrow \epsilon_1 (m_2 = k_{\text{max}}/4-1/2, k_2(k'))$ conductor $\text{Ext}_2$ with $k_2(k') = (k' - 3\log(m_2/\epsilon_1) - 1)/\rho_2$ for $k' \in [0, k_{\text{max}}/2 - 1]$. If we take these two conductors and concatenate them together, we get according to Lemma 3.21 (with $s = 1$) a third extracting conductor $\text{Ext}_3 = \text{Ext}_1 \| \text{Ext}_2$ which is an $(n, 0, k_{\text{max}}) \times (d_1 + d_1) \rightarrow \epsilon_3 (m_3 = 3 \cdot k_{\text{max}}/4 - 1/2, k_3(k') = k' - \Delta_3(k'))$ conductor with maximal entropy loss $\Delta_3(k_{\text{max}}) = k_{\text{max}}/4 + 1/2$ and $\epsilon_3 = 3\epsilon_1$. This can be seen by the following: For the output length, we have $m_3 = m_1 + m_2 = k_{\text{max}}/2 + k_{\text{max}}/4 - 1/2 = 3 \cdot k_{\text{max}}/4 - 1/2$, for the maximal entropy loss $\Delta_3(k_{\text{max}}) = \Delta_2(k_{\text{max}}/2 - 1) + 1 = (k_{\text{max}}/2 - 1)/2 + 1 = k_{\text{max}}/4 + 1/2$ and for the error $\epsilon_3 = 2 \cdot \epsilon_1 + \epsilon_1 = 3\epsilon_1$. Now, we take again a Trevisan extracting conductor which extracts up to $\Delta_3(k_{\text{max}}) - 1$ min-entropy as second conductor in the concatenation and concatenate it with $\text{Ext}_3$. We do this iterated concatenation until we get at step $j$ an entropy loss of $\Delta_j(k_{\text{max}}) = \log(k_{\text{max}})$.

The question is now how many iteration steps we need to reduce the entropy loss to $\log(k_{\text{max}})$ because at each iteration we resize the seed by the summand $d$. By analyzing the iterated concatenation, we see that at step $i$ we use as second conductor an $(n, 0, \Delta_{i-1}(k_{\text{max}}) - 1) \times$
(d) \rightarrow_{\epsilon_1} (m_i, k_i(k')) conductor with
\[
\Delta_{i-1}(k_{\text{max}}) - 1 = \frac{k_{\text{max}}}{2^{i-1}} - \frac{1}{2^{i-2}} \quad m_i = \frac{k_{\text{max}}}{2^i} - \frac{1}{2^{i-1}}
\]
Thus, for the number of needed iteration steps \(j\) we have
\[
\frac{k_{\text{max}}}{2^j} - \frac{1}{2^{j-1}} + 1 = \log(k_{\text{max}})
\]
\[
2^j = \frac{k_{\text{max}} - 2}{\log(k_{\text{max}}) - 1}
\]
\[
j = \log\left(\frac{k_{\text{max}} - 2}{\log(k_{\text{max}}) - 1}\right).
\]
Note that if at step \(j\) we get \(\Delta_j(k_{\text{max}}) < \log(k_{\text{max}})\) we can remove some bits from \(m_j\) to get a loss of \(\log(k_{\text{max}})\).

For our approximations of the final error \(\epsilon\) and the final seed length \(d = j \cdot d_1\), we bound \(j \leq \log(k_{\text{max}})\) and hence we get
\[
\epsilon \leq (2^{\log(k_{\text{max}})} - 1) \cdot \epsilon_1 = (k_{\text{max}} - 1) \cdot \epsilon_1
\]
and
\[
\log(k_{\text{max}}) \cdot d_1 = \log(k_{\text{max}}) \cdot \left[\frac{\ell}{\ln(\rho)}\right] \cdot \ell \leq \psi(k_{\text{max}}, \epsilon) \cdot \ell^2 \cdot \log(k_{\text{max}}) + \ell \cdot \log(k_{\text{max}})
\]
where \(\psi\) and \(\ell\) are as defined in Section 4.1.3.

Summarizing, we constructed the following conductor:

**Theorem 5.1.** For \(k_{\text{max}} \leq n\) and a constant \(2^{\log(k_{\text{max}}) + \log(k_{\text{max}} - 1) - k_{\text{max}}/6 + 2/3} < \epsilon < 1\), there exist a strong injective extracting \((n, 0, k_{\text{max}}) \times (d) \rightarrow_{\epsilon} (m, k_m(k'))\) conductor Ext\(_T\) with \(k_m(k') = k' - \log(k')\) for \(k' \in [0, k_{\text{max}}]\) and seed length \(d = \psi(k_{\text{max}}, \epsilon) \cdot \ell^2 \cdot \log(k_{\text{max}}) + \ell \cdot \log(k_{\text{max}})\).

### 5.2 Extracting Conductor with Almost Optimal Entropy Loss

In this section, we present a strong extracting conductor which is almost ideal in respect of its entropy loss and which will be used as a building block in the construction of Section 5.3. The construction of this section was originally discussed in [RRV99, MST04]. We present a more detailed construction and give additionally the exact values of the conductor parameters. The general idea of getting such an almost ideal extracting conductor is to first apply a strong extracting conductor Ext\(_1\) which works for big min-entropy, e.g. \(k > \log n\). The output string will then contain \(k \rightarrow \Delta_1\) min-entropy. But this means that \(\Delta_1\) min-entropy of the input is yet not extracted. Therefore, we will apply a second strong extracting conductor Ext\(_2\) which works well for small min-entropies to extract most of the remaining \(\Delta_1\) min-entropy. Overall, this leads to a much smaller entropy loss \(\Delta_2\) then if we would just have applied the first conductor Ext\(_1\).
5.2. EXTRACTING CONDUCTOR WITH ALMOST OPTIMAL ENTROPY LOSS

We present now a possible choice for Ext₁ and Ext₂ such that their concatenation according to Lemma 3.21 leads to a strong injective extracting \((n, 0, k_{\text{max}}) \times (d) \to \epsilon (m, k' \mapsto k' - \Delta)\) conductor with almost ideal constant entropy loss \(\Delta\) and \(k' \in [0, k_{\text{max}}]\). See Figure 5.1 for an illustration of the construction.

**Choosing First Conductor.** We instantiate \(\text{Ext}_1 : \{0,1\}^n \times \{0,1\}^{d_T} \to \{0,1\}^{m_T}\) with the strong injective extracting \((n, 0, k_{\text{max}}) \times (d_T) \to \epsilon (m_T, k_T(k'))\) iterated Trevisan conductor with entropy loss \(\Delta_T(k')\) from Section 5.1 for \(k' \in [0, k_{\text{max}}]\) such that:

- \(\epsilon_1 = \epsilon_T/(k_{\text{max}} - 1)\)
- \(m_T = k_{\text{max}} - \log(k_{\text{max}})\)
- \(k_T(k') = k' - \log(k')\)
- \(\Delta_T(k') = \log(k')\)
- \(\epsilon_T = \epsilon/4\)
- \(d_T = \psi \cdot \ell^2 \cdot \log(k_{\text{max}}) + \ell \cdot \log(k_{\text{max}}),\) where we used \(\psi\) as short-hand for \(\psi(k_{\text{max}}, \epsilon_1)\)

**Choosing Second Conductor.** We know that there are \(\Delta_T(k')\) bits of the \(n\)-bit input string which we have not yet extracted. Therefore, for the second strong extracting conductor \(\text{Ext}_2\), we use the same \(n\)-bit input string but set now the maximal min-entropy to \(\Delta_T(k_{\text{max}}) - 1\). We instantiate \(\text{Ext}_2 : \{0,1\}^n \times \{0,1\}^{d_{\text{SZ}}} \to \{0,1\}^{m_{\text{SZ}}}\) with the extracting \((n, 0, k_{\text{SZ}}) \times (d_{\text{SZ}}) \to \epsilon_{\text{SZ}} (m_{\text{SZ}}, k' \mapsto k' - \Delta_{\text{SZ}})\) conductor from Section 4.2 as following:

- \(k_{\text{SZ}} = \Delta_T(k_{\text{max}}) - 1 = \log(k_{\text{max}}) - 1\)
- \(\epsilon_{\text{SZ}} = \epsilon/2\)
- \(\Delta_{\text{SZ}} = 2 \log(2/\epsilon) + 2 = 2 \log(1/\epsilon) + 4\)
- \(m_{\text{SZ}} = k_{\text{SZ}} - \Delta_{\text{SZ}} = \log(k_{\text{max}}) - 2 \log(1/\epsilon) - 5\)
- \(d_{\text{SZ}} = 6 \log(n) + 4k_{\text{SZ}} = 6 \log(n) + 4 \log(k_{\text{max}}) - 4\)

Concatenating \(\text{Ext}_1\) with \(\text{Ext}_2\) according to Lemma 3.21 (setting \(s = 1\)) gives the following strong injective extracting conductor \(\text{Ext}_{RRV}\) with \(d_{\text{RRV}} = d_T + d_{\text{SZ}}\) and entropy loss \(\Delta_{\text{RRV}} = \Delta_{\text{SZ}} + s\).

**Theorem 5.2.** For every constant \(2^{-\log(k_{\text{max}})+\log(k_{\text{max}}-1)-k_{\text{max}}/6+8/3} < \epsilon < 1\) and \(k_{\text{max}} \leq n\), \(\text{Ext}_{RRV} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m\) is a strong injective extracting \((n, 0, k_{\text{max}}) \times (d) \to \epsilon (m, k_m(k'))\) conductor with \(k_m(k') = k' - \Delta\) for \(\Delta = 2 \log(1/\epsilon) + 5\) and \(k' \in [0, k_{\text{max}}]\) and seed length \(d = \psi(k_{\text{max}}, \epsilon) \cdot \ell^2 \cdot \log(k_{\text{max}}) + \ell \cdot \log(k_{\text{max}}) + 6 \log(n) + 4 \log(k_{\text{max}}) - 4\).
\[ \Delta = \Delta_{SZ} + 1 = 2 \log(1/\epsilon) + 5 \]

\[ \epsilon = 2 \cdot \epsilon_T + \epsilon_{SZ} \]

Figure 5.1: Construction of an extracting conductor with almost optimal entropy loss

**A Possible Value for Seed Length \( d \).** If we use the EC code from Section 4.1.3, we have

\[ \ell \approx 6 \log n + 4 \log(1/\epsilon) \]

and hence, for \( d \) we get

\[
d = \psi \cdot (6 \log(n) + 4 \log(1/\epsilon))^2 \cdot \log(k_{max}) + (6 \log(n) + 4 \log(1/\epsilon)) \cdot \log(k_{max}) \\
+ 6 \log(n) + 4 \log(k_{max}) - 4 \\
= 36 \cdot \psi \cdot \log^2(n) \cdot \log(k_{max}) + (6 + 48 \cdot \psi \cdot \log(1/\epsilon)) \cdot \log(n) \cdot \log(k_{max}) \\
+ 6 \log(n) + ((16 \cdot \psi + 4) \cdot \log(1/\epsilon) + 4) \cdot \log(k_{max}) - 4
\]

We will use this instantiation of \( Ext_{RRV} \) later in Section 5.4.

### 5.3 Improved Conductor by First Condensing

In this section, we introduce a strong injective extracting conductor \( Ext_{MST} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^n \) due to [MST04] and give its concrete parameters. It is an improvement of the conductor presented in the previous Section 5.2. The main idea is to first apply a strong injective and condensing conductor to the input \( x \in \{0,1\}^n \) to preserve the min-entropy but to reduce the string length to \( n' \) and second, apply a strong injective extracting conductor with almost ideal entropy loss. As condensing conductor, we use \( C_{TUZ} \) from Section 4.3.2 and as extracting conductor, we choose \( Ext_{RRV} \) from Section 5.2. Given the condensing
5.3. IMPROVED CONDUCTOR BY FIRST CONDENSING

Conductor $C_{TUZ}$ and the extracting conductor $Ext_{RRV}$, we can describe the extracting conductor $Ext_{MST}$ as follows. We interpret input $y \in \{0,1\}^d$ of $Ext_{MST}$ as a concatenation of two strings $y_1$ and $y_2$ where $y_1 \in \{0,1\}^{d_{TUZ}}$ is used for the condensing conductor $C_{TUZ}$ and $y_2 \in \{0,1\}^{d_{RRV}}$ is used for the extracting conductor $Ext_{RRV}$. By applying Lemma 3.19, the conductor $Ext_{MST} : \{0,1\}^n \times \{0,1\}^d \rightarrow \{0,1\}^m$ is defined as

$$Ext_{MST} := Ext_{RRV}(C_{TUZ}(x,y_1),y_2).$$

The overall construction is illustrated in Figure 5.2. Note that because $C_{TUZ}$ and $Ext_{RRV}$ are both injective, the final conductor $Ext_{MST}$ will be injective, too.

![Figure 5.2: Conductor Ext_{MST}](image)

We instantiate now the different parameters of the conductors such that we get a conductor with almost optimal min-entropy as the conductor of Section 5.2 but needs a shorter seed.

For the EC algorithm used in the construction of $Ext_{RRV}$ (for the Trevisan subpart), we use the EC encoding introduced in Section 4.1.3 and set $EC' : \{0,1\}^n' \rightarrow \{0,1\}^{n'}$ for $Ext_{RRV}$. In the construction of $C_{TUZ}$ we need an EC encoding with relative distance $> 0.1$. We will assume such an encoding being of the form $EC : \{0,1\}^n \rightarrow \{0,1\}^{n'}$ for a constant $s^1$. We fix the constants $\epsilon$ and $\delta$, choose a $k_{max} \leq n$ and we see that the total length of the seed is $d = d_{TUZ} + d_{RRV}$.

For the condensing $(n,0,k_{TUZ}) \times (d_{TUZ}) \rightarrow (m_{TUZ},k' \mapsto k')$ conductor $C_{TUZ}$ we set

- $\epsilon_{TUZ} = \epsilon/8$ (we will show why)
- $k_{TUZ} = k_{max}$
- $\ell = \log(n^s) = s \log(n)$
- $m_{TUZ} = n' = (k_{max} + \ell/\epsilon_{TUZ})^{1+2\delta}$
- $d_{TUZ} = (2 + \delta) \cdot \ell/\alpha$.

And for the extracting $(n',0,k_{RRV}) \times (d_{RRV}) \rightarrow (m_{RRV},k' \mapsto k' - \Delta_{RRV})$ conductor $Ext_{RRV}$ we set

- $\epsilon_{RRV} = 5 \cdot \epsilon/8$ (we will show why)

\(^1\)For example, the encoding of Section 4.1.3 has this form and relative distance of more than 0.1 if we set $\sigma = 2$.
• \(k_{RRV} = k_{\text{max}}\)
• \(l' = \log(n')\)
• \(\Delta_{RRV} = \lceil 2 \log(1/\epsilon_{RRV}) + 5 \rceil = 2 \log(1/\epsilon) + 7\)
• \(d_{RRV} = \psi \cdot (l')^2 \cdot \log(k_{\text{max}}) + \ell \cdot \log(k_{\text{max}}) + 6 \log(n') + 4 \log(k_{\text{max}}) - 4\)
• \(m_{RRV} = k_{\text{max}} - \Delta_{RRV}\)

Note that the output length \(n'\) can be longer than \(n\) if \(k_{\text{max}}\) is getting too big. But this is no problem as we will see later at the calculation for the needed seed length.

First, we analyze why we have chosen \(\epsilon_{TUZ} = \epsilon/8\) and \(\epsilon_{RRV} = 5\epsilon/8\). We cannot choose \(\epsilon_{TUZ} = \epsilon_{RRV} = \epsilon/2\) as for the normal conductor composition, because \(Ext_{RRV}\) uses the \(n'\)-bit string with error \(\epsilon_{TUZ}\) twice, once for calling \(Ext_T\) with internal error \(\epsilon_{Ti}\) from Section 4.1 and once for calling \(C_{SZ}\) with internal error \(\epsilon_{SZi}\) from Section 4.2. As in Figure 5.3 illustrated, we see that by setting \(\epsilon_{Ti} = \epsilon/8\) and \(\epsilon_{SZi} = 3 \cdot \epsilon/8\) we get for the overall error \(\epsilon_{MST}\)

\[
\epsilon_{MST} = 2 \cdot (\epsilon_{TUZ} + \epsilon_{Ti}) + (\epsilon_{TUZ} + \epsilon_{SZi}) \\
= 2 \cdot (\epsilon/8 + \epsilon/8) + \epsilon/8 + 3 \cdot \epsilon/8 = \epsilon,
\]

as wished. Hence, \(\epsilon_{RRV} = 2 \cdot \epsilon_{Ti} + \epsilon_{SZi} = 5 \cdot \epsilon/8\).

\[
\Delta = \Delta_{SZ} + 1 = 2 \log(1/\epsilon_{RRV}) + 5 \leq 2 \log(1/\epsilon) + 7 \quad \epsilon = 2 \cdot \epsilon_T + \epsilon_{SZ}
\]

Figure 5.3: Detailed construction
5.3. IMPROVED CONDUCTOR BY FIRST CONDENSING

We will give now the calculations for the different $d$'s. We start with the value of $d_{TUZ}$.

$$d_{TUZ} = \frac{2 + \delta}{\alpha} \cdot \ell = \frac{s \cdot (2 + \delta)}{\alpha} \cdot \log(n).$$

We calculate now the value of $d_{RRV}$ step by step:

$$d_{RRV} = \psi \cdot (\ell')^2 \cdot \log(k_{max}) + \ell' + 6 \log(n') + 4 \log(k_{max} - 4) =: d_T.$$

We know that

$$n' = \left( \frac{8 \cdot (k_{max} + \ell/\alpha)}{\epsilon} \right)^{1+2\delta}.$$

This gives for the value of $\ell'$:

$$\ell' \leq 6 \log(n') + 4 \log(1/\epsilon_{TI}) = 6 \log(n') + 4 \log(1/\epsilon) + 12$$

with

$$6 \log(n') = 6 \cdot (1 + 2\delta) \cdot \log(k_{max} + \ell/\alpha) + 6 \cdot (1 + 2\delta) \log 8 + 6 \cdot (1 + 2\delta) \log(1/\epsilon)$$

$$= 6 \cdot (1 + 2\delta) \cdot \log(k_{max} + \ell/\alpha) + 6 \cdot (1 + 2\delta) \log(1/\epsilon) + 18 \cdot (1 + 2\delta).$$

(1)

Furthermore, we have $\ell = s \cdot \log(n)$. This gives

$$\log(k_{max} + \ell/\alpha) = \log \left( k_{max} + \frac{s}{\alpha} \cdot \log(n) \right)$$

$$\leq \log \left( k_{max} + \frac{s}{\alpha} k_{max} \right)$$

$$= \log (k_{max}) + \log \left( 1 + \frac{s}{\alpha} \right).$$

(2)

where at $*$ we assumed that the upper bound $k_{max}$ is greater than $\log n$.

Inserting (2) into (1) gives

$$6 \log(n') \leq 6(1 + 2\delta) \cdot \log(k_{max}) + 6(1 + 2\delta) \cdot (\log (1 + s/\alpha) + \log(1/\epsilon) + 2),$$

and hence, for the value of $\ell'$,

$$\ell' \leq 6(1 + 2\delta) \cdot \log(k_{max}) + 6(1 + 2\delta) \cdot (\log (1 + s/\alpha) + \log(1/\epsilon) + 2) + 4 \log(1/\epsilon) + 12.$$

We collect now all constants to a new constant $\varphi$

$$\varphi := 6(1 + 2\delta) \cdot (\log (1 + s/\alpha) + \log(1/\epsilon) + 2) + 4 \log(1/\epsilon) + 12$$

hence,

$$\ell' \leq 6(1 + 2\delta) \cdot \log(k_{max}) + \varphi$$

and

$$6 \log(n') \leq 6(1 + 2\delta) \cdot \log(k_{max}) + \varphi - 4 \log(1/\epsilon) - 12.$$

We get

$$d_T = \psi \cdot (\ell')^2 \cdot \log(k_{max}) + \ell' \leq \psi \cdot (6(1 + 2\delta) \log(k_{max}) + \varphi)^2 \cdot \log(k_{max}) + 6(1 + 2\delta) \log(k_{max}) + \varphi$$

$$= 36 \cdot \psi \cdot (1 + 2\delta)^2 \cdot \log^2(k_{max}) + 12 \cdot \psi \cdot \varphi (1 + 2\delta) \cdot \log^2(k_{max})$$

$$+ \psi \cdot \varphi^2 \cdot \log(k) + (\psi \cdot \varphi^2 + 6(1 + 2\delta)) \cdot \log(k_{max}) + \varphi$$
and
\[
d_{SZ} = 6 \log(n') + 4 \log(k_{\text{max}}) - 4 \leq 6(1 + 2\delta) \cdot \log(k_{\text{max}}) + \varphi - 4 \log(1/\epsilon) - 12
\]
\[+ 4 \log(k_{\text{max}}) - 4
\]
\[= (6(1 + 2\delta) + 4) \cdot \log(k_{\text{max}}) + \varphi - 4 \log(1/\epsilon) - 16
\]
\[\leq (6(1 + 2\delta) + 4) \cdot \log(k_{\text{max}}) + \varphi,
\]
where at the last step we used \(4 \log(1/\epsilon) > 0\). In total, we get for \(d_{RRV}\)
\[
d_{RRV} = \psi \cdot (\ell')^2 \cdot \log(k_{\text{max}}) + \ell' + 6 \log(n') + 4 \log(k_{\text{max}}) - 4
\]
\[\leq 36 \cdot \psi \cdot (1 + 2\delta)^2 \cdot \log^3(k_{\text{max}}) + 12 \cdot \psi \cdot \varphi \cdot (1 + 2\delta) \cdot \log^2(k_{\text{max}})
\]
\[+ (\psi \cdot \varphi^2 + 12(1 + 2\delta) + 4) \cdot \log(k_{\text{max}}) + 2\varphi.
\]
Finally, we have for the total seed length needed for the \(Ext_{MST}\) conductor:
\[
d_{MST} = d_{TUZ} + d_{RRV}
\]
\[\leq \frac{s \cdot (2 + \delta)}{\alpha} \cdot \log(n) + 36 \cdot \psi \cdot (1 + 2\delta)^2 \cdot \log^3(k_{\text{max}})
\]
\[+ 12 \cdot \psi \cdot \varphi \cdot (1 + 2\delta) \cdot \log^2(k_{\text{max}}) + (\psi \cdot \varphi^2 + 12(1 + 2\delta) + 4) \cdot \log(k_{\text{max}}) + 2\varphi
\]
\[\in O(\log n + \log^3(k_{\text{max}})).
\]
Summarizing, we get the following strong injective extracting conductor \(Ext_{MST}\), which needs seed length \(O(\log n + \log^3(k_{\text{max}}))\) instead of \(O(\log^3 n)\) if we had just used \(Ext_{RRV}\). As long as \(k\) is much smaller than \(n\) this is an improvement.

**Theorem 5.3.** Let \(EC : \{0,1\}^n \rightarrow \{0,1\}^{n''}\) be an EC code with relative distance \(> 0.1\). For every \(k_{\text{max}} \leq n\) and every constants \(\epsilon, \delta > 0\), there exists an explicit and injective strong extracting \((n,0,k_{\text{max}}) \times (d) \rightarrow \epsilon (m,k_m(k'))\) conductor \(Ext_{MST}\) with \(k_m(k') = k' - \Delta\) for \(k' \in [0,k_{\text{max}}]\) and \(\Delta = 2 \cdot \log(1/\epsilon) + 7\) and seed length
\[
d = d(n,k_{\text{max}},\epsilon,\delta)
\]
\[= \frac{s \cdot (2 + \delta)}{\alpha} \cdot \log(n) + 36 \cdot \psi \cdot (1 + 2\delta)^2 \cdot \log^3(k_{\text{max}})
\]
\[+ 12 \cdot \psi \cdot \varphi \cdot (1 + 2\delta) \cdot \log^2(k_{\text{max}}) + (\psi \cdot \varphi^2 + 12(1 + 2\delta) + 4) \cdot \log(k_{\text{max}}) + 2\varphi
\]
where \(\varphi\) is a constant with the value
\[
\varphi = 6(1 + 2\delta) \cdot (\log (1 + s/\alpha) + \log(1/\epsilon) + 2) + 4 \log(1/\epsilon) + 12
\]
and \(\alpha\) a constant with
\[
\alpha = \frac{\delta}{2 + \delta} \cdot \frac{1}{s \cdot \log \epsilon}.
\]

**Conductor from Theorem 5.3 as Expander Graph.** From Theorem 3.27, we know that this explicit injective strong extracting conductor is an expander graph with left-degree \(2^d\). Unfortunately, this expander graph cannot be used for the application introduced in Section 1.2 because there, we want to have \(k_{\text{max}} \in \Theta(n)\) which leads to a super-polynomial left-degree in \(n\), but we want to have an expander graph with polynomial left-degree. Therefore, we will introduce in the next section a construction of a somewhere-conductor based on this conductor which will lead to an expander graph with polynomial bounded left-degree.
5.4 Final Explicit Construction of an Injective Unbalanced Bipartite Expander Graph

In this section, we give a concrete construction of an expander graph fulfilling the requirements to be applicable for the domain extension of public random functions introduced in Section 1.2. In particular, the expander graph will have a polynomial-bounded left-degree and $K_{\text{max}} = 2^{k_{\text{max}}}$ will be in the order of $2^{\Theta(n)}$. We present an adapted construction due to [MT07, BJST03] and give detailed values to show that the left-degree is $\text{poly}(n)$ even if $K_{\text{max}} = 2^{\Theta(n)}$.

We use the somewhere conductor construction of Section 3.3.3 and we instantiate $C_1$ by applying Theorem 5.3 and $C_2$ with the conductor from Theorem 5.2. Note that in [MT07] both conductors $C_1$ and $C_2$ where instantiated with the conductor of Theorem 5.3. Our construction gives asymptotically the same results but the construction is simpler. For $C_2$, we set the maximal input min-entropy for $k_2 = k_{\text{max}} = (1 - \eta)n$ for some constant $\eta$ and set the maximal input min-entropy for $C_1$ to $k_1 = d_2 + \Delta$ for a constant $\Delta$. Further, we set $\nu = 3 \cdot \log(9n)$ for having $\sigma < 1$, fix the constant $\epsilon$, set $a_1 = a_2 = -\Delta = -2 \log(1/\epsilon) - 5$ and have $m = k_2 + d_2 - \Delta$.

Hence, for $d_2$, we get

$$d_2 = d(n, k_2 = (1 - \eta)n, \epsilon) = 36 \cdot \psi \cdot \log^2(n) \cdot \log((1 - \eta)n) + (6 + 48 \cdot \psi \cdot \log(1/\epsilon)) \cdot \log(n) \cdot \log((1 - \eta)n) + 6 \log(n) + (((16 \cdot \psi + 4) \cdot \log(1/\epsilon) + 4) \cdot \log((1 - \eta)n) - 4 \in O(\log^3(n))$$

and for $d_1$ we get:

$$d_1 = d(n, k_1 = d_2 + \Delta, \epsilon, \delta) = \frac{s \cdot (2 + \delta)}{\alpha} \cdot \log(n) + 36 \cdot \psi \cdot (1 + 2\delta)^2 \cdot \log^3(d_2) + 12 \cdot \psi \cdot \varphi \cdot (1 + 2\delta) \cdot \log^2(d_2) + (\psi \cdot \varphi^2 + 12(1 + 2\delta) + 4) \cdot \log(d_2) + 2\varphi \in O(\log(n))$$

where $\varphi$ and $\alpha$ are constants with the value

$$\varphi = 6(1 + 2\delta) \cdot (\log(1 + s/\alpha) + \log(1/\epsilon) + 2) + 4 \log(1/\epsilon) + 12$$

and $\psi$ is the function described in Theorem 5.2. Further, we have

$$C_1 : \{0, 1\}^n \times \{0, 1\}^{d_1} \rightarrow \{0, 1\}^{d_2}, \quad k_1 = d_2 + \Delta$$
$$C_2 : \{0, 1\}^n \times \{0, 1\}^{d_2} \rightarrow \{0, 1\}^m, \quad k_2 = k = (1 - \eta)n$$

and therefore

$$C : \{0, 1\}^n \times \{0, 1\}^{d_1} \rightarrow \{0, 1\}^{n \cdot (d_1 + d_2 + m)}$$

is a $\sigma$-somewhere $(n, 0, k_{\text{max}}) \times (d_1) \rightarrow_{\text{2r}} (n \cdot (d_1 + d_2 + m), k_m(k'))$ conductor with $k_m(k') = k' + d_1 - 2\Delta - \nu$ and $\sigma < 1$ and its construction needs only $d_1 = O(\log n)$ truly random bits.

To get the final expander graph, we can apply Lemma 3.29 to the above somewhere conductor. We see that the left-degree $D$ of the expander is $n \cdot 2^{d_1}$, for the expander parameter
Chapter 5. Compositions of Basic Constructions

$K_{\text{max}}$ we have $K_{\text{max}} = 2^{6\text{max}} = 2^{(1-\eta)n}$ and the expansion factor is $\gamma = 2^{d_1 - 2\Delta - \nu} \cdot (1 - 2\epsilon)$. Note that the final expander graph is injective because all the conductors used to compose the somewhere conductor are injective conductors.

If we take a closer look at the value of $d_1$, we see that the first summand $s \cdot (2+\delta) \cdot \log n$ specifies the highest degree of the polynomial $D(n) = n \cdot 2^{d_1}$. Thus, $d_1 = \frac{s \cdot (2+\delta) \cdot \log(n)}{\alpha} \log n + f(n)$ for a function $f$ being in $o(\log(n))$. In particular, the highest degree of $D(n)$ is not dependent on the chosen error $\epsilon$, the constant $\eta$ and the expansion factor $\gamma$.

**Estimation for the Highest Degree of the Polynomial $D(n)$:** We have $\frac{s \cdot (2+\delta) \cdot \log(n)}{\alpha} = s^2 \cdot \log e \cdot \frac{2+\delta^2}{\alpha} \cdot \log n$ for $\alpha = \frac{\delta}{2+\delta} \cdot \frac{1}{s \log e}$ and an EC encoding $\{0,1\}^n \to \{0,1\}^n$ with relative distance $> 0.1$ used in the construction of $C_1$. Let us now choose a concrete EC encoding which satisfies the requirements for the construction of $C_1$.

We take the EC code used in Section 4.1.3 and set its parameter $\sigma = 2$ to get a relative distance of $1/4$. Thus, we get $\ell \approx 4n^2$ and hence, $s \approx 2$. Furthermore, $\frac{(2+\delta)^2}{\alpha}$ is minimal for $\delta = 2$. We get $s^2 \cdot \log e \cdot \frac{(2+\delta)^2}{\alpha} \cdot \log n \geq 8 \cdot 4^2 \log e \cdot \log n \approx 184.5 \cdot \log n$. For the left-degree $D$, we have:

$$D \geq n \cdot 2^{128 \cdot \log(e) \cdot \log(n)} \approx n^{185.5}.$$ 

Thus, for practice it is not really applicable to use the above construction of an expander graph. But there is still a potential to further improve the degree of the polynomial $D(n)$, in particular, the right choice of the EC encoding plays a big role. We know that the value of $s$ is lower bounded by 1. Hence, for a more compact EC encoding than the one in Section 4.1.3, we would get $s^2 \cdot \log e \cdot \frac{(2+\delta)^2}{\alpha} \cdot \log n \geq 8 \cdot \log e \cdot \log n \approx 11.5 \cdot \log n$ and hence

$$D \geq n \cdot 2^{8.1 \cdot \log(e) \cdot \log(n)} \approx n^{12.5}$$

which is still not practicable but much better than what can be achieved with the EC code of Section 4.1.3. Unfortunately, we did not have time left to find an EC code with relative distance $> 0.1$ and having an encoding length $\bar{n}$ such that $\log(\bar{n}) = (1 + \xi) \cdot \log(n)$ for a small constant $\xi \ll 1$. Nevertheless, even with $s = 1$, we get $\Omega(n^{12.5})$ for the left-degree. Thus, one would need a strong condensing conductor which has shorter seed length than the conductor of Section 4.3. But to our knowledge, there is no strong condensing conductor construction with shorter seed than the conductor of Section 4.3.

Overall, we showed that

**Lemma 5.4.** For every polynomially-bounded $\gamma$ and every constant $\eta \in (0,1)$, and all functions $m$ (polynomially-bounded in $n$), there exists an explicit injective family of $(2^n,0,K) \times (D) \rightarrow (2^m)$ expander graphs $G = (V_1, V_2, E)$ with $K = 2^n(1-\eta)$ and left-degree $D$ polynomially-bounded in $n$. 

72
6 Graph Construction with Substring Selection

The previous chapters introduced quite complex constructions of expander graphs using conductors and the final expander graph construction in Section 5.4 has a left-degree size which is not really practicable. In this chapter, we analyze a candidate for expander graph constructions which is simple and would lead to small left-degree. The idea is to select $D$ substrings of length $n$ from a longer string of length $rn$ and interpret the substrings as neighbors of the longer string. In particular, we get a graph $G = (V_1, V_2, E)$ with $V_1 = \{0, 1\}^{rn}$ and $V_2 = \{0, 1\}^n$, where $r$ is an integer and the graph would have a left-degree of $D$. We show that any such construction based on substring selection cannot lead to a $(2^{rn}, K_{\min}, K_{\max}) \times (D) \rightarrow (2^n)$ expander graph with useful expansion factor $\gamma$ if we require to allow a big $K_{\max} \in 2^{\Theta(n)}$.

6.1 Candidate for Expander Graph Construction

We start by a simple instantiation of the described idea of substring selection. We choose $r = 2$ and $D = 2$. We select the two $n$-bit neighbors as follows: For every left vertex $v \in \{0, 1\}^{2n}$, we split its $2n$-bit representation into two $n$-bits, where the first $n$ bits of the representation is interpreted as the first neighbor of vertex $v$ and the second $n$-bits represent the second neighbor of $v$. An illustration is given in Figure 6.1.

![Figure 6.1: Simple graph construction](image)

Before we analyze what expansion factor we can hope for, we denote a formal notion for the "selection function". Let $X = \{x_1, x_2, ..., x_m\}$ be a set of strings $x_i \in \{0, 1\}^{2n}$. Further let $\mathcal{P} = \{P_1, P_2, ..., P_\ell\}$ be a family of projections $P_i \subset [2n]$ with $|P_i| = n$ for all $i$'s. $P_i$ denotes which $n$ bits should be taken from a string $x_j$ for the $i$th neighbor and we denote $S_{i|P_i} := \bigcup_{j=1}^{m} x_{j|P_i}$. Hence, the $i$th neighbor of $x$ would be $x_{|P_i}$ and we have $\Gamma(X) := \bigcup_{i=1}^{\ell} X_{|P_i}$.

For our simple instantiation of substring selection, we have $\ell = 2$, $\mathcal{P} = \{P_1, P_2\}$ with $P_1 = \{1, 2, ..., n\}$ and $P_2 = \{n + 1, n + 2, ..., 2n\}$. To give a non-trivial upper bound for the expansion factor $\gamma$, we choose $\sqrt{K_{\max}}$ distinct $n$-bit strings and denote them as the neighbor set $\Gamma(X)$. We see now that if we take all possible pairwise combinations of this $\sqrt{K_{\max}}$ distinct $n$-bit strings, we get $\sqrt{K_{\max}}^2 = K_{\max}$ distinct $2n$-bit strings which is our set $X$. Thus, the expansion factor is at most $\gamma \leq \frac{\Gamma(X)}{\alpha} = \frac{\sqrt{K_{\max}}}{K_{\max}} = \frac{1}{\sqrt{K_{\max}}}$. If we want to have a $K_{\max} \in 2^{\Theta(n)}$ as required in the application of expander graphs described in Section 3.6, we would get an
exponentially small expansion factor $\gamma \in 2^{-\Omega(n)}$. Such an expansion factor is not usable, in particular not for the application in Section 3.6.

**Lemma 6.1.** There exists a set $\mathcal{X}$ with $|\mathcal{X}| = m$ and $m \leq 2^n$ such that $\gamma \leq \frac{1}{\sqrt[4]{m}}$ and $P = \{P_1, P_2\}$ with $P_1 = \{1, \ldots, n\}$ and $P_1 = \{n + 1, \ldots, 2n\}$.

In the next section, we show that even having a longer input string of length $r \cdot n$ for an $r > 1$ and allowing to choose the $n$ bits arbitrarily for each neighbor and to choose more than two substrings, we still get an exponentially small expansion factor in $n$ and hence, the graph construction based on substring selection does not give usable expander graphs. In particular, we show the following theorem.

**Theorem 6.2.** Let $r > 1$ and $K \in 2^{\Theta(n)}$ be constants. Then there exists no family $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ of projections $P_i \subset [rn]$ with $|P_i| = n$ for all $i$’s and an integer $\ell$, which would lead to an expander graph with upper bound $K_{\max} = K$ and not having exponentially small expansion factor $\gamma$ in $n$.

### 6.2 Impossibility Proof

In this section, we show that increasing the number of neighbors for each vertex and allowing selecting the $n$-bit substrings arbitrary does not improve the construction. We still get expander graphs with exponentially small expansion factor. To show a non-trivial upper bound for the expansion factor $\gamma$, we assume that $\mathcal{V}_1 = \{0, 1\}^{rn}$ for an integer $r > 1$ and define a special set $\mathcal{X} \subset \{0, 1\}^{rn}$ of the $rn$-bit strings such that this set has only few neighbors. We instantiate $\mathcal{X}$ as being the set of all strings in $\{0, 1\}^{rn}$ with maximal $k$ bits set to 1. We define $k = \phi \cdot rn$ for a constant $\phi$ such that $k \leq n$. We restrict $k$ to $n$ because the number of neighbors cannot be bigger than $2^n$. Furthermore, we have $r \geq 2$ and hence, $\phi \leq 1/2$.

The number of all possible $rn$-bit strings with maximal $k$ 1s is at least

$$|\mathcal{X}| = \sum_{i=0}^{k} \binom{rn}{i} = \sum_{i=0}^{\phi \cdot rn} \binom{rn}{i} \geq \binom{rn}{\phi \cdot rn} \geq \frac{2^{rn-h(\phi)}}{e \cdot \sqrt{2\pi(1-\phi)rn}},$$

where at the last step, we applied Lemma 2.21 and $h(\cdot)$ is the binary entropy function.

Now, to estimate the expansion factor, we need an upper bound for the number of possible neighbors $\Gamma(\mathcal{X})$ for set $\mathcal{X}$. We know that every neighbor of $\mathcal{X}$ can only have at most $k$ bits set to 1 because at most $k$ 1 bits are contained in the string representation of every vertex in $\mathcal{X}$. Let $\mathcal{M}$ be the set of $n$-bit strings with maximal $k$ bits set to 1, thus, we have $|\Gamma(\mathcal{X})| \leq |\mathcal{M}|$.

To estimate the size of $\mathcal{M}$, we distinguish three cases:

**Case 1.** $r \cdot \phi \leq 1/2$

We can apply Lemma 2.20 for bounding:

$$|\mathcal{M}| = \sum_{i=0}^{\phi \cdot rn} \binom{n}{i} \leq 2^{n-h(\phi r)}$$

Hence, we get the following upper bound for the expansion factor $\gamma$:

$$\gamma \leq \frac{|\Gamma(\mathcal{X})|}{|\mathcal{X}|} \leq \frac{|\mathcal{M}|}{|\mathcal{X}|} \leq \sqrt{2\pi(1-\phi)rn} \cdot e^{\frac{2^{rn-h(\phi r)}}{2rn-h(\phi)}} \leq \sqrt{2\pi(1-\phi)rn} \cdot e^{\frac{2^{n-h(\phi r-h(\phi))}}{2rn-h(\phi)}}.$$
Hence, we get

\[ \gamma \in 2^{-\Omega(n)}. \]

**Case 2.** \( r\phi > 1/2 \) and \( r\phi \neq 1 \)

This time, we have to bound \( |\mathcal{M}| \) differently, because Lemma 2.20 is not directly applicable. For the size of \( \mathcal{M} \), we count the cases where the number of 1s is up to \( n/2 \) and add the number of cases where the number of 0s is between \( n - k = n - \phi rn \) and \( n/2 - 1 \). This leads to

\[
|\mathcal{M}| = \sum_{i=0}^{n/2} \binom{n}{i} + \left( \sum_{i=0}^{n/2-1} \binom{n}{i} - \sum_{i=0}^{n-\phi rn} \binom{n}{i} \right)
\]

\[
= 2 \cdot \sum_{i=0}^{n/2} \binom{n}{i} - \binom{n}{n/2} - \sum_{i=0}^{(1-\phi)n} \binom{n}{i}
\]

\[
\leq 2 \cdot \sum_{i=0}^{n/2} \binom{n}{i} - (1-\phi)n \cdot \sum_{i=0}^{n} \binom{n}{i}
\]

\[
\leq 2 \cdot 2^{(n-1)} - 2^n \cdot 2^{h(1-\phi)n)} = 2^n \cdot (1 - 2^{h(1-\phi)}).
\]

Hence, we get

\[
\gamma \leq \frac{\Gamma(\mathcal{X})}{|\mathcal{X}|} \leq \frac{|\mathcal{M}|}{|\mathcal{X}|} \leq \sqrt{2\pi \phi (1 - \phi) rn \cdot e} \cdot \frac{2^n \cdot (1 - 2^{h(1-\phi)})}{2^{rn \cdot h(\phi)}},
\]

where we used the fact that \( \sum_{i=0}^{n/2} \binom{n}{i} = \sum_{i=n/2}^{n} \binom{n}{i} \) and applied again Lemma 2.20. We still have \( h(r\phi) < r \cdot h(\phi) \) and because \( r\phi > 1/2 \), we have \( r \cdot h(\phi) > 1 \). Thus, the expansion factor is as in Case 1 in \( 2^{-\Omega(n)} \).

**Case 3.** \( r\phi = 1 \)

Finally, we regard the last possible case, where \( r\phi = 1 \), which we omitted in Case 2, because it would have lead to the wrong statement \( |\mathcal{M}| \leq 0 \).

We have \( k = r\phi n = n \) and hence, the set \( \mathcal{M} \) is just the set of all strings in \( \{0,1\}^n \). Thus,

\[
|\mathcal{M}| = 2^n.
\]

Therefore, we get for the expansion factor \( \gamma \):

\[
\gamma \leq \frac{\Gamma(\mathcal{X})}{|\mathcal{X}|} \leq \frac{|\mathcal{M}|}{|\mathcal{X}|} \leq \sqrt{2\pi \phi (1 - \phi) rn \cdot e} \cdot \frac{2^n}{2^{rn \cdot h(\phi)}},
\]

As in Case 2, we have \( r \cdot h(\phi) > 1 \) and hence, the expansion factor \( \gamma \) is again exponentially small in \( n \).

Overall, we get for all three possible cases that \( \gamma \in 2^{-\Omega(n)} \) if we try to construct an expander graph with the construction idea of selecting \( n \)-bit substrings and allowing \( K_{\text{max}} \) to be of the order \( 2^{\Theta(n)} \). Hence, this simple constructions are not useful for the application described in Section 1.2. Although, we could achieve a small left-degree, we get a bad expansion factor.
7 Conclusions and Outlook

In this thesis, we analyzed an expander graph construction which to our knowledge is the best construction known regarding the size of the left-degree and we calculated the concrete value of the left-degree. Although, the left-degree is small in complexity-theoretic terms, its concrete value is too big for the expander graph construction being of practical interest. There are possible improvements for the size of the left-degree, which we did not analyze. For example, by finding a better error-correcting code, used in the construction, which has the needed properties but gives shorter encodings, or by finding a strong condensing conductor for the conductor construction in Section 5.3, which has a shorter seed length than the used strong condensing conductor due to [TUZ01].

Another parameter of the expander graph construction, which we did not investigate in, but could be of interest, is the concrete running time complexity of the expander graph function calculating the neighbors. In particular, the constructions of Sections 4.1 and 4.3 use weak designs as basic constructs and what time complexity their construction has could be interesting.

In Chapter 6, we analyzed an alternative construction of expander graphs which does not rely on conductors. Unfortunately, this alternative construction based on substring selection does not lead to useful expander graphs. But because expander graphs have in general weaker properties than conductors, it is possible that a simple construction not relying on conductors would give expander graphs with much smaller left-degree. One possible construction, which could be further considered is a construction based on linear transformation: Assume that we want to construct an unbalanced bipartite expander graph $G = (V_1, V_2, E)$ with $|V_1| = 2^{rn}$, $|V_2| = 2^n$ and left-degree $D$, where $r > 1$, then we could see the function calculating the neighbors as a function $F^r \times F^D \rightarrow F$ where $F$ is the field $GF(2^n)$. Or describing as matrix operation: Given a matrix $A \in F^{D \times r}$ and a vector $v \in F^r$ being a vertex $v \in V_1$, the neighbors of $v$ are $A \cdot v$. The question is now what kind of matrices $A$ leads to a good expansion parameter. Unfortunately, we did not have the time to further investigate in this construction proposition, but it seems to be a promising construction.
Bibliography


Bibliography


List of Figures

1.1 Construction of public random function $\{0, 1\}^n \rightarrow \{0, 1\}^\ell$ .......................... 2

2.1 Binary entropy function .......................... 6

3.1 Example for an expander graph: $K_{5,3}$ .................................................. 13
3.2 Function $C$ extracting $p$ random bits from a $k$-source ............................................. 16
3.3 Conductor cascading .................................................. 18
3.4 Conductor concatenation .................................................. 19
3.5 Construction of $C(i)(x,y)$ .................................................. 23

4.1 Function family $G_D$ .................................................. 46

5.1 Construction of an extracting conductor with almost optimal entropy loss ...... 66
5.2 Conductor $Ext_{MST}$ .................................................. 67
5.3 Detailed construction .................................................. 68

6.1 Simple graph construction .................................................. 73
List of Figures
Symbols
\( \epsilon \)-close, 7
\( t \)-wise \( \rho \)-biased sample space, 50
conductor, 15
  cascading, 18
  concatenation, 19, 21
  condensing, 16
  explicit, 15
  extracting, 16
  injective, 15
  lossless, 16
  reconstructive extracting, 53
  somewhere, 17
  strong, 15
A
advantage, see distinguishing advantage
advice function, 54
advice string, 42
B
binary entropy function, see entropy
binomial coefficient, 9
bipartite graph, 10
  balanced, 10
  unbalanced, 14
C
Cauchy-Schwarz inequality, 7
Chernoff bound, 6, 30
composition theorems, 18
condenser, 39
conductor, 39
D
degree, 11
design, 42
distinguisher, 6
distinguishing advantage, 6
E
entropy, 5
  binary entropy function, 5
entropy loss, 16
expander graph, 13, 14
  explicit, 14
  injective, 14
  unbalanced, 13
explicit, 14
extractor, 39
F
flat distribution, 6
G
generalized conductor, see conductor
generalized expander graph, see expander graph
graph, 10
I
injective, 14
input-restricting function family, 3, 37
K
k-source, 7
L
left-degree, 11
leftover hash lemma, 50
M
min-entropy, 5
multi graph, 10
N
next-bit predictor, 8
Nisan-Widgerson generator, 41
R
reconstructive extracting conductor, see conductor
Index

right-degree, 11

S
seed, 15
somewhere conductor, see conductor
source, see k-source
statistical difference, 6
support, 6, 14

T
Trevisan's construction, 41

W
weak design, 42
A Task Description

Concrete Constructions of Unbalanced Bipartite Expander Graphs and Generalized Conductors

Master’s Project for Rose-Line Werner


A.1 Introduction

A $(K, \gamma)$-bipartite expander graph $G = (V_1, V_2, E)$ is a bipartite graph\(^1\) with the following additional property: For all subsets $X \subseteq V_1$ such that $|X| \leq K$, the cardinality of the set $\Gamma(X) \subseteq V_2$ of all neighbors of the vertices in $X$ satisfies $|\Gamma(X)| \geq \gamma |X|$. Such a graph is called unbalanced when $|V_1| > |V_2|$. In general, one is interested in the case where both $V_1$ and $V_2$ are large, i.e. exponential in some given security parameter $n$. In order for such a graph to be useful, there must exist an algorithm which, given a vertex $v \in V_1$ and an index $i$, can compute efficiently (that is, in time polynomial in $n$) the $i$’th neighbor of $v$.

Interestingly, it turns out that unbalanced bipartite expander graphs play an important role in cryptography. In particular, they have been used in the following two contexts.

**Bounded-Storage Model.** In the bounded storage model [Mau92, CM97] security is proved under the sole assumption that an adversary has bounded storage capabilities, but is otherwise computationally unbounded. In this model, a long random string $R$ is initially broadcast to all parties (both the honest parties and the adversary), but the adversary’s memory resources do not allow him to store all of $R$. The honest parties interact in order to perform some cryptographic task (e.g. generating a secret key) depending on $R$, and security has to be guaranteed as long as the adversary can only store a portion of it, even if he has much more memory than the honest parties.

Moran, Shaltiel, and Ta-Shma [MST04] have considered the problem of non-interactive timestamping of documents in the bounded-storage model: The idea is that honest users can timestamp documents, but the adversary is not able to produce more timestamps than what his memory would allow him to do in the case he would behave honestly. Their solution relies on the use of unbalanced bipartite expander graphs in order to select which random bits of $R$ are used to timestamp a message.

**Extension of Public Random Primitives.** A well-known problem in cryptography is the problem of extending random resources in a secure way. For example, one would like to obtain a longer secret key from a given shorter secret key. While this problem is well-understood

\(^1\)That is, all edges $\{u, v\} \in E$ are between a vertex $u \in V_1$ and a vertex $v \in V_2$. 
in the setting where randomness is private, it is less clear when honest users would like to extend randomness which is public, i.e. also accessible by the adversary. An example of such a primitive is a public random function that maps $m$-bit strings to $n$-bit strings, i.e. a system which takes an $m$-bit string as input (both from the honest parties and form the adversary), and for each such input returns consistently a uniformly-distributed $n$-bit string.

In recent work [MT07], we have presented a solution for extending a public random function mapping $n$-bit string to $n$-bit string to a public random function mapping arbitrarily-long bit strings to $n$-bit strings, and which guarantees nearly-optimal security. The approach we take is a novel one, and relies on the use of unbalanced expander graphs.

A.2 Description

Despite the potential wide range of possible applications of unbalanced bipartite expander graphs to cryptography (and to computer science in general), their study has been rather limited so far. Such graphs have only been constructed from much stronger objects (for example, randomness extractors [NT99]), and the existing constructions achieve parameters which are even too strong for many applications, at the cost of an inherent inefficiency.

The goal of this thesis is to provide an overview of unbalanced bipartite expander graphs, their properties, and their applications to information-theoretic cryptography. In particular, in the first part of the work, the student is supposed to gain an overview on existing results on unbalanced bipartite expander graphs and to present this in a survey providing a unifying view on the topic.

Furthermore, it would be interesting to study possible graph constructions, and to study their expanding properties. In fact, there are some natural candidates for sufficiently-good constructions using only basic mathematics, but it is not obvious to decide whether their expanding properties are good enough to obtain parameters suitable for cryptographic applications.

Finally, in light of the results of [MST04] and [MT07], it is interesting to understand the role of the expansion property of such graphs in these results, and to apply similar ideas to other tasks in information-theoretic cryptography. It may also be possible that slightly weaker tools suffice to obtain cryptographic applications.

A.3 Tasks

Possible tasks for this project are the following ones.

1. Study the relevant literature on expander graphs (and in particular on unbalanced bipartite expander graphs) and formulate a clear survey of existing results.

2. By using known composition theorems, study compositions of known constructions and the parameters that can be achieved, including the parameters $K$ and $\gamma$, as well as good estimates of the sizes of the resulting graphs.

3. Study direct constructions (based for example on simple algebraic or number theoretic ideas) of unbalanced bipartite graphs as well as their expanding properties. Are they good expanders? And if not, what is the reason?
4. Work on [MST04, MT07] and try to exploit the acquired insights to understand the essence of these results. In particular, which properties and which parameters are the relevant ones? Would it be possible to weaken the concept of unbalanced bipartite expander graphs and still achieve the same results?

5. Possibly, try to find new cryptographic tasks where the use of unbalanced bipartite expander graphs could lead to a novel solution.

By the end of the project the work shall be presented in a talk. Hints about the documentation can be found in the enclosed guidelines.

Requirements for this work are interest in theoretical research as well as good mathematical-thinking skills. Only basic knowledge in cryptography is assumed (as in the basic Cryptography lecture).

A.4 Grading of the Thesis

The master's project encompasses independent scientific research, writing a Master's thesis, and giving a presentation. The evaluation of the thesis takes into account the quality of the results (understanding of the subject, contributed ideas, correctness) and the quality of the documentation (thesis and presentation). More instructions for the documentation and information about grading criteria can be found in the enclosed leaflets.