Applications of Derandomization Theory in Coding

by

Mahdi Cheraghchi Bashi Astaneh

Master of Science (École Polytechnique Fédérale de Lausanne), 2005

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

at the

School of Computer and Communication Sciences

École Polytechnique Fédérale de Lausanne

Thesis Number: 4767

Committee in charge:

Emre Telatar, Professor (President)
Amin Shokrollahi, Professor (Thesis Director)
Rüdiger Urbanke, Professor
Venkatesan Guruswami, Associate Professor
Christopher Umans, Associate Professor

July 2010

Applications of Derandomization Theory in Coding

Abstract

Randomized techniques play a fundamental role in theoretical computer science and discrete mathematics, in particular for the design of efficient algorithms and construction of combinatorial objects. The basic goal in derandomization theory is to eliminate or reduce the need for randomness in such randomized constructions. Towards this goal, numerous fundamental notions have been developed to provide a unified framework for approaching various derandomization problems and to improve our general understanding of the power of randomness in computation. Two important classes of such tools are *pseudorandom generators* and *randomness extractors*. Pseudorandom generators transform a short, purely random, sequence into a much longer sequence that *looks* random, while extractors transform a weak source of randomness into a perfectly random one (or one with much better qualities, in which case the transformation is called a *randomness condenser*).

In this thesis, we explore some applications of the fundamental notions in derandomization theory to problems outside the core of theoretical computer science, and in particular, certain problems related to coding theory. First, we consider the *wiretap channel problem* which involves a communication system in which an intruder can eavesdrop a limited portion of the transmissions. We utilize randomness extractors to construct efficient and information-theoretically optimal communication protocols for this model.

Then we consider the *combinatorial group testing* problem. In this classical problem, one aims to determine a set of defective items within a large population by asking a number of queries, where each query reveals whether a defective item is present within a specified group of items. We use randomness condensers to explicitly construct optimal, or nearly optimal, group testing schemes for a setting where the query outcomes can be highly unreliable, as well as the *threshold model* where a query returns positive if the number of defectives pass a certain threshold.

Next, we use randomness condensers and extractors to design ensembles of error-correcting codes that achieve the information-theoretic capacity of a large class of communication channels, and then use the obtained ensembles for construction of explicit capacity achieving codes. Finally, we consider the problem of explicit construction of error-correcting codes on the *Gilbert-Varshamov bound* and extend the original idea of Nisan and Wigderson to obtain a small ensemble of codes, mostly achieving the bound, under suitable computational hardness assumptions.

Keywords: Derandomization theory, randomness extractors, pseudorandomness, wiretap channels, group testing, error-correcting codes.

Résumé

Les techniques de randomisation jouent un rôle fondamental en informatique théorique et en mathématiques discrètes, en particulier pour la conception d'algorithmes efficaces et pour la construction d'objets combinatoires. L'objectif principal de la théorie de dérandomisation est d'éliminer ou de réduire le besoin d'aléa pour de telles constructions. Dans ce but, de nombreuses notions fondamentales ont été développées, d'une part pour créer un cadre unifié pour aborder différents problèmes de dérandomisation, et d'autre part pour mieux comprendre l'apport de l'aléa en informatique. Les *générateurs pseudo-aléatoires* et les *extracteurs* sont deux classes importantes de tels outils. Les générateurs pseudo-aléatoires transforment une suite courte et purement aléatoire en une suite beaucoup plus longue qui *parait* aléatoire. Les *extracteurs d'aléa* transforment une source faiblement aléatoire en une source parfaitement aléatoire (ou en une source de meilleure qualité. Dans ce dernier cas, la transformation est appelée un *condenseur d'aléa*).

Dans cette thèse, nous explorons quelques applications des notions fondamentales de la théorie de dérandomisation à des problèmes périphériques à l'informatique théorique et en particulier à certains problèmes relevant de la théorie des codes. Nous nous intéressons d'abord au *problème du canal à jarretière*, qui consiste en un système de communication où un intrus peut intercepter une portion limitée des transmissions. Nous utilisons des extracteurs pour construire pour ce modèle des protocoles de communication efficaces et optimaux du point de vue de la théorie de l'information.

Nous étudions ensuite le problème du *test en groupe combinatoire*. Dans ce problème classique, on se propose de déterminer un ensemble d'objets défectueux parmi une large population, à travers un certain nombre de questions, où chaque réponse révèle si un objet défectueux appartient à un certain ensemble d'objets. Nous utilisons des condenseurs pour construire explicitement des tests de groupe optimaux ou quasi-optimaux, dans un contexte où les réponses aux questions peuvent être très peu fiables, et dans le *modèle de seuil* où le résultat d'une question est positif si le nombre d'objets défectueux dépasse un certain seuil.

Ensuite, nous utilisons des condenseurs et des extracteurs pour concevoir des ensembles de codes correcteurs d'erreurs qui atteignent la capacité (dans le sens de la théorie de l'information) d'un grand nombre de canaux de communications. Puis, nous utilisons les ensembles obtenus pour la construction de codes explicites qui atteignent la capacité. Nous nous intéressons finalement au problème de la construction explicite de codes correcteurs d'erreurs qui atteignent la *borne de Gilbert–Varshamov* et reprenons l'idée originale de Nisan et Wigderson pour obtenir un petit ensemble de codes dont la plupart atteignent la borne, sous certaines *hypothèses de difficulté computationnelle.*

Mots-clés: Théorie de dérandomisation, extracteurs d'aléa, pseudo-aléa, canaux à jarretière, test en groupe, codes correcteurs d'erreurs.

Acknowledgments

During my several years of study at EPFL, both as a Master's student and a Ph.D. student, I have had the privilege of interacting with so many wonderful colleagues and friends who have been greatly influential in my graduate life. Despite being thousands of miles away from home, thanks to them my graduate studies turned out to be one of the best experiences of my life. These few paragraphs are an attempt to express my deepest gratitude to all those who made such an exciting experience possible.

My foremost gratitude goes to my adviser, Amin Shokrollahi, for not only making my academic experience at EPFL truly enjoyable, but also for numerous other reasons. Being not only a great adviser and an amazingly brilliant researcher but also a great friend, Amin is undoubtedly one of the most influential people in my life. Over the years, he has taught me more than I could ever imagine. Beyond his valuable technical advice on research problems, he has thought me how to be an effective, patient, and confident researcher. He would always insist on picking research problems that are worth thinking, thinking about problems for the joy of thinking and without worrying about the end results, and publishing only those results that are worth publishing. His mastery in a vast range of areas, from pure mathematics to engineering real-world solutions, has always greatly inspired for me to try learning about as many topics as I can and interacting with people with different perspectives and interests. I'm especially thankful to Amin for being constantly available for discussions that would always lead to new ideas, thoughts, and insights. Moreover, our habitual outside-work discussions in restaurants, on the way for trips, and during outdoor activities turned out to be a great source of inspiration for many of our research projects, and in fact some of the results presented in this thesis! I'm also grateful to Amin for his collaborations on several research papers that we coauthored, as well as the technical substance of this thesis. Finally I thank him for numerous small things, like encouraging me to buy a car which turned out to be a great idea!

Secondly, I would like to thank our secretary Natascha Fontana for being so patient with too many inconveniences that I made for her over the years! She was about the first person I met in Switzerland, and kindly helped me settle in Lausanne and get used to my new life there. For several years I have been constantly bugging her with problems ranging from administrative trouble with the doctoral school to finding the right place to buy curtains. She has also been a great source of encouragement and support for my graduate studies.

Besides Amin and Natascha, I'm grateful to the present and past members of our Laboratory of Algorithms (ALGO) and Laboratory of Algorithmic Mathematics (LMA) for creating a truly active and enjoyable atmosphere:

Bertrand Meyer, Ghid Maatouk, Giovanni Cangiani, Harm Cronie, Hesam Salavati, Luoming Zhang, Masoud Alipour, Raj Kumar (present members), and Andrew Brown, Bertrand Ndzana Ndzana, Christina Fragouli, Frédéric Didier, Frédérique Oggier, Lorenz Minder, Mehdi Molkaraie, Payam Pakzad, Pooya Pakzad, Zeno Crivelli (past members), as well as Alex Vardy, Emina Soljanin, Martin Fürer, and Shahram Yousefi (long-term visitors). Special thanks to:

Alex Vardy and Emina Soljanin: For fruitful discussions on the results presented in Chapter 3.

Frédéric Didier: For numerous fruitful discussions and his collaboration on our joint paper [32], on which Chapter 3 is based.

Giovanni Cangiani: For being a brilliant system administrator (along with Damir Laurenzi), and his great help with some technical problems that I had over the years.

Ghid Maatouk: For her lively presence as an endless source of fun in the lab, for taking student projects with me prior to joining the lab, helping me keep up my obsession about classical music, encouraging me to practice the piano, and above all, being an amazing friend. I also thank her and Bertrand Meyer for translating the abstract of my thesis into French.

Lorenz Minder: For sharing many tech-savvy ideas and giving me a quick campus tour when I visited him for a day in Berkeley, among other things.

Payam Pakzad: For many fun activities and exciting discussions we had during the few years he was with us in ALGO.

Zeno Crivelli and Bertrand Ndzana Ndzana: For sharing their offices with me for several years! I also thank Zeno for countless geeky discussions, lots of fun we had in the office, and for bringing a small plant to the office, which quickly grew to reach the ceiling and stayed fresh for the entire duration of my Ph.D. work.

I'm thankful to professors and instructors from whom I learned a great deal attending their courses as a part of my Ph.D. work: I learned Network Information Theory from Emre Telatar, Quantum Information Theory from Nicolas Macris, Algebraic Number Theory from Eva Bayer, Network Coding from Christina Fragouli, Wireless Communication from Suhas Diggavi, and Modern Coding Theory from my adviser Amin. As a teaching assistant, I also learned a lot from Amin's courses (on algorithms and coding theory) and from an exciting collaboration with Monika Henzinger for her course on advanced algorithms.

During summer 2009, I spent an internship at KTH working with Johan Håstad and his group. What I learned from Johan within this short time turned out far more than I had expected. He was always available for discussions and listening to my countless silly ideas with extreme patience, and I would always walk out of his office with new ideas (ideas that would, contrary to those of my own, always work!). Working with the theory group at KTH was more than enjoyable, and I'm particularly thankful to Ola Svensson, Marcus Isaksson, Per Austrin, Cenny Wenner, and Lukáš Poláček for numerous delightful discussions.

Special thanks to cool fellows from the Information Processing Group (IPG) of EPFL for the fun time we had and also countless games of Foosball we played (brought to us by Giovanni).

I'm indebted to my great friend, Soheil Mohajer, for his close friendship over the years. Soheil has always been patient enough to answer my countless questions on information theory and communication systems in a computer-science-friendly language, and his brilliant mind has never failed to impress me. We had tons of interesting discussions on virtually any topic, some of which coincidentally (and finally!) contributed to a joint paper [34]. I also thank Amin Karbasi and Venkatesh Saligrama for this work. Additional thanks goes to Amin for our other joint paper [33] (along with Ali Hormati and Martin Vetterli whom I also thank) and in particular giving me the initial motivation to work on these projects, plus his unique sense of humor and amazing friendship over the years.

I'd like to extend my warmest gratitude to Pedram Pedarsani, for too many reasons to list here, but above all for being an amazingly caring and supportive friend and making my graduate life even more pleasing. Same goes to Mahdi Jafari, who has been a great friend of mine since middle school! Mahdi's many qualities, including his humility, great mind, and perspective to life (not to mention great photography skills) has been a big influence on me. I feel extremely lucky for having such amazing friends.

I take this opportunity to thank three of my best, most brilliant, and most influential friends; Omid Etesami, Mohammad Mahmoody, and Ehsan Ardestanizadeh, whom I'm privileged to know since high school. In college, Omid showed me some beauties of complexity theory which strongly influenced me in pursuing my post-graduate studies in theoretical computer science. He was also influential in my decision to study at EPFL, which turned out to be one of my best decisions in life. I had the most fascinating time with Mohammad and Ehsan during their summer internships at EPFL. Mohammad thought me a great deal about his fascinating work on foundations of cryptography and complexity theory and was always up to discuss anything ranging from research ideas to classical music and cinema. I worked with Ehsan on our joint paper [6] which turned out to be one of the most delightful research collaborations I've had. Ehsan's unique personality, great wit and sense of humor, as well as musical talents—especially his mastery in playing Santur—

has always filled me with awe. I also thank the three of them for keeping me company and showing me around during my visits in Berkeley, Princeton, and San Diego.

In addition to those mentioned above, I'm grateful to so many amazing friends who made my study in Switzerland an unforgettable stage of my life and full of memorable moments: Ali Ajdari Rad, Amin Jafarian, Arash Golnam, Arash Salarian, Atefeh Mashatan, Banafsheh Abasahl, Elham Ghadiri, Faezeh Malakouti, Fereshteh Bagherimiyab, Ghazale Hosseinabadi, Hamed Alavi, Hossein Afshari, Hossein Rouhani, Hossein Taghavi, Javad Ebrahimi, Laleh Golestanirad, Mani Bastani Parizi, Marjan Hamedani, Marjan Sedighi, Maryam Javanmardy, Maryam Zaheri, Mina Karzand, Mohammad Karzand, Mona Mahmoudi, Morteza Zadimoghaddam, Nasibeh Pouransari, Neda Salamati, Nooshin Hadadi, Parisa Haghani, Pooyan Abouzar, Pouya Dehghani, Ramtin Pedarsani, Sara Kherad Pajouh, Shirin Saeedi, Vahid Aref, Vahid Majidzadeh, Wojciech Galuba, and Zahra Sinaei. Each name should have been accompanied by a story (ranging from a few lines to a few pages); however, doing so would have made this section exceedingly long. Moreover, having prepared the list rather hastily, I'm sure I have missed a lot of nice friends on it. I owe them a coffee (or tea, if they prefer) each! Additional thanks to Mani, Ramtin, and Pedram, for their musical presence.

Thanks to Alon Orlitsky, Avi Wigderson, Madhu Sudan, Rob Calderbank, and Umesh Vazirani for arranging my short visits to UCSD, IAS, MIT, Princeton, and U.C. Berkeley, and to Anup Rao, Swastik Kopparty, and Zeev Dvir for interesting discussions during those visits.

I'm indebted to Chris Umans, Emre Telatar, Rüdiger Urbanke, and Venkat Guruswami for giving me the honor of having them in my dissertation committee. I also thank them (and Amin) for carefully reading the thesis and their comments on an earlier draft of this work. Additionally, thanks to Venkat for numerous illuminating discussions on various occasions, in particular on my papers [30, 31] that form the basis of the material presented in Chapter 4.

My work was in part funded by grants from the Swiss National Science Foundation (Grant No. 200020-115983/1) and the European Research Council (Advanced Grant No. 228021) that I gratefully acknowledge.

Above all, I express my heartfelt gratitude to my parents, sister Azadeh, and brother Babak who filled my life with joy and happiness. Without their love, support, and patience none of my achievements—in particular this thesis—would have been possible. I especially thank my mother for her everlasting love, for all she went through until I reached this point, and her tremendous patience during my years of absence while I was only able to go home for a short visit each year. This thesis is dedicated with love to her.

Contents

List of Figures

List of Tables

"You are at the wheel of your car, waiting at a traffic light, you take the book out of the bag, rip off the transparent wrapping, start reading the first lines. A storm of honking breaks over you; the light is green, you're blocking traffic."

— *Italo Calvino*

Chapter 1

Introduction

Over the decades, the role of randomness in computation has proved to be one of the most intriguing subjects of study in computer science. Considered as a fundamental computational resource, randomness has been extensively used as an indispensable tool in design and analysis of algorithms, combinatorial constructions, cryptography, and computational complexity.

As an illustrative example on the power of randomness in algorithms, consider a *clustering* problem, in which we wish to partition a collection of items into two groups. Suppose that some pairs of items are marked as *inconsistent*, meaning that they are best be avoided falling in the same group. Of course, it might be simply impossible to group the items in such a way that no inconsistencies occur within the two groups. For that reason, it makes sense to consider the objective of minimizing the number of inconsistencies induced by the chosen partitioning. Suppose that we are asked to color individual items red or blue, where the items marked by the same color form each of the two groups. How can we design a strategy that maximizes the number of inconsistent pairs that fall in different groups? The basic rule of thumb in randomized algorithm design suggests that

When unsure making decisions, try flipping coins!

Thus a naive strategy for assigning color to items would be to flip a fair coin for each item. If the coin falls Heads, we mark the item blue, and otherwise red.

How can the above strategy possibly be any reasonable? After all we are defining the groups without giving the slightest thought on the given structure of the inconsistent pairs! Remarkably, a simple analysis can prove that the coin-flipping strategy is in fact a quite reasonable one. To see why, consider any inconsistent pair. The chance that the two items are assigned the same color is exactly one half. Thus, we expect that half of the inconsistent pairs

end up falling in different groups. By repeating the algorithm a few times and checking the outcomes, we can be sure that an assignment satisfying half of the inconsistency constraints is found after a few trials.

We see that, a remarkably simple algorithm that does not even read its input can attain an approximate solution to the clustering problem in which the number of inconsistent pairs assigned to different groups is no less than half the maximum possible. However, our algorithm used a valuable resource; namely random coin flips, that greatly simplified its task. In this case, it is not hard to come up with an efficient (i.e., polynomial-time) algorithm that does equally well without using any randomness. However, designing such an algorithm and analyzing its performance is admittedly a substantially more difficult task that what we demonstrated within a few paragraphs above.

As it turns out, finding an optimal solution to our clustering problem above is an intractable problem (in technical terms, it is NP-hard), and even obtaining an approximation ratio better than $16/17 \approx .941$ is so [79]. Thus the trivial bit-flipping algorithm indeed obtains a reasonable solution. In a celebrated work, Goemans and Williamson [69] improve the approximation ratio to about .878, again using randomization[1]. A deterministic algorithm achieving the same quality was later discovered [104], though it is much more complicated to analyze.

Another interesting example demonstrating the power of randomness in algorithms is the *primality testing* problem, in which the goal is to decide whether a given n-digit integer is prime or composite. While efficient (polynomial-time in n) randomized algorithms were discovered for this problem as early as 1970's (e.g., Solovay-Strassen's [140] and Miller-Rabin's algorithms [107, 121]), a deterministic polynomial-time algorithm for primality testing was found decades later, with the breakthrough work of Agrawal, Kayal, and Saxena [3], first published in 2002. Even though this algorithm provably works in polynomial time, randomized methods still tend to be more favorable and more efficient for practical applications.

The primality testing algorithm of Agrawal et al. can be regarded as a derandomization of a particular instance of the *polynomial identity testing* problem. Polynomial identity testing generalizes the high-school-favorite problem of verifying whether a pair of polynomials expressed as closed form formulae expand to identical polynomials. For example, the following is an 8-variate identity

$$\begin{aligned}(x_1^2+x_2^2+x_3^2+x_4^2)(y_1^2+y_2^2+y_3^2+y_4^2) \stackrel{?}{\equiv} \\ (x_1y_1-x_2y_2-x_3y_3-x_4y_4)^2+(x_1y_2+x_2y_1+x_3y_4-x_4y_3)^2+ \\ (x_1y_3-x_2y_4+x_3y_1+x_4y_2)^2+(x_1y_4+x_2y_3-x_3y_2+x_4y_1)^2\end{aligned}$$

[1] Improving upon the approximation ration obtained by this algorithm turns out to be NP-hard under a well-known conjecture [90].

which turns out to be valid. When the number of variables and the complexity of the expressions grow, the task of verifying identities becomes much more challenging using naive methods.

This is where the power of randomness comes into play again. A fundamental idea due to Schwartz and Zippel [131, 169] shows that the following approach indeed works:

> *Evaluate the two polynomials at sufficiently many randomly chosen points, and identify them as identical if and only if all evaluations agree.*

It turns out that the above simple idea leads to a randomized efficient algorithm for testing identities that may err with an arbitrarily small probability. Despite substantial progress, to this date no polynomial-time deterministic algorithms for solving general identity testing problem is known, and a full *derandomization* of Schwartz-Zippel's algorithm remains a challenging open problem in theoretical computer science.

The discussion above, among many other examples, makes the strange power of randomness evident. Namely, in certain circumstances the power of randomness makes algorithms more efficient, or simpler to design and analyze. Moreover, it is not yet clear how to perform certain computational tasks (e.g., testing for general polynomial identities) without using randomness.

Apart from algorithms, randomness has been used as a fundamental tool in various other areas, a notable example being combinatorial constructions. Combinatorial objects are of fundamental significance for a vast range of theoretical and practical problems. Often solving a practical problem (e.g., a real-world optimization problem) reduces to construction of suitable combinatorial objects that capture the inherent structure of the problem. Examples of such combinatorial objects include graphs, set systems, codes, designs, matrices, or even sets of integers. For these constructions, one has a certain structural property of the combinatorial object in mind (e.g., mutual intersections of a set system consisting of subsets of a universe) and seeks for an instance of the object that optimizes the property in mind in the best possible way (e.g., the largest possible set system with bounded mutual intersections).

The task of constructing suitable combinatorial objects turns out quite challenging at times. Remarkably, in numerous situations the power of randomness greatly simplifies the task of constructing the ideal object. A powerful technique in combinatorics, dubbed as *the probabilistic method* (see [5]) is based on the following idea:

> *When out of ideas finding the right combinatorial object, try a random one!*

Surprisingly, in many cases this seemingly naive strategy significantly beats the most brilliant constructions that do not use any randomness. An illuminating example is the problem of constructing *Ramsey graphs*. It is well known

that in a group of six or more people, either there are at least three people who know each other or three who do not know each other. More generally, *Ramsey theory* shows that for every positive integer K, there is an integer N such that in a group of N or more people, either there are at least K people who mutually know each other (called a *clique* of size K) or K who are mutually unfamiliar with one another (called an *independent set* of size K). Ramsey graphs capture the reverse direction:

> *For a given N, what is the smallest K such that there is a group of N people with no cliques or independent sets of size K or more? And how can an example of such a group be constructed?*

In graph-theoretic terms (where mutual acquaintances are captured by edges), an undirected graph with $N := 2^n$ vertices is called a Ramsey graph with *entropy* k if it has no clique or independent set of size $K := 2^k$ (or larger). The Ramsey graph construction problem is to efficiently construct a graph with smallest possible entropy k.

Constructing a Ramsey graph with entropy $k = (n+1)/2$ is already non-trivial. However, the following *Hadamard graph* does the job [35]: Each vertex of the graph is associated with a binary vector of length n, and there is an edge between two vertices if their corresponding vectors are orthogonal over the binary field. A much more involved construction, due to Barak et al. [9] (which remains the best deterministic construction to date) attain an entropy $k = n^{o(1)}$.

A brilliant, but quite simple, idea due to Erdős [57] demonstrates the power of randomness in combinatorial constructions: Construct the graph randomly, by deciding whether to put an edge between every pair of vertices by flipping a fair coin. It is easy to see that the resulting graph is, with overwhelming probability, a Ramsey graph with entropy $k = \log n + 2$. It also turns out that this is about the lowest entropy one can hope for! Note the significant gap between what achieved by a simple, probabilistic construction versus what achieved by the best known deterministic constructions.

Even though the examples discussed above clearly demonstrate the power of randomness in algorithm design and combinatorics, a few issues are inherently tied with the use of randomness as a computational resource, that may seem unfavorable:

1. A randomized algorithm takes an abundance of fair, and independent, coin flips for granted, and the analysis may fall apart if this assumption is violated. For example, in the clustering example above, if the coin flips are biased or correlated, the .5 approximation ratio can no longer be guaranteed. This raises a fundamental question:

Does "pure randomness" even exist? If so, how can we instruct a computer program to produce purely random coin flips?

2. Even though the error probability of randomized algorithms (such as the primality testing algorithms mentioned above) can be made arbitrarily small, it remains nonzero. In certain cases where a randomized algorithm never errs, its running time may vary depending on the random choices being made. We can never be completely sure whether an error-prone algorithm has really produced the right outcome, or whether one with a varying running time is going to terminate in a reasonable amount of time (even though we can be almost confident that it does).

3. As we saw for Ramsey graphs, the probabilistic method is a powerful tool in showing that combinatorial objects with certain properties exist, and it most cases it additionally shows that a random object almost surely achieves the desired properties. Even though for certain applications a randomly produced object is good enough, in general there might be no easy way to certify whether a it indeed satisfies the properties sought for. For the example of Ramsey graphs, while almost every graph is a Ramsey graph with a logarithmically small entropy, it is not clear how to certify whether a given graph satisfies this property. This might be an issue for certain applications, when an object with *guaranteed* properties is needed.

The basic goal of *derandomization theory* is to address the above-mentioned and similar issues in a systematic way. A central question in derandomization theory deals with efficient ways of *simulating randomness*, or relying on *weak randomness* when perfect randomness (i.e., a steady stream of fair and independent coin flips) is not available. A mathematical formulation of randomness is captured by the notion of *entropy*, introduced by Shannon [136], that quantifies randomness as the amount of *uncertainty* in the outcome of a process. Various sources of "unpredictable" phenomena can be found in nature. This can be in form of an electric noise, thermal noise, ambient sound input, image captured by a video camera, or even a user's input given to an input device such as a keyboard. Even though it is conceivable to assume that a bit-sequence generated by all such sources contains a certain amount of entropy, the randomness being offered might be far from perfect. Randomness *extractors* are fundamental combinatorial, as well as computational, objects that aim to address this issue.

As an example to illustrate the concept of extractors, suppose that we have obtained several independent bit-streams $X_1, X_2, \ldots, X_r$ from various physically random sources. Being obtained from physical sources, not much is known about the structure of these sources, and the only assumption that we can be confident about is that they produce a substantial amount of entropy.

An extractor is a function that combines these sources into one, *perfectly random*, source. In symbols, we have

$$f(X_1, X_2, \ldots, X_r) = Y,$$

where the output source Y is purely random provided that the input sources are reasonably (but not fully) random. To be of any practical use, the extractor f must be efficiently computable as well. A more general class of functions, dubbed *condensers* are those that do not necessarily transform imperfect randomness into perfect one, but nevertheless substantially purifies the randomness being given. For instance, as a condenser, the function f may be expected to produce an output sequence whose entropy is 90% of the optimal entropy offered by perfect randomness.

Intuitively, there is a trade-off between *structure* and *randomness*. A sequence of fair coin flips is extremely unpredictable in that one cannot *bet* on predicting the next coin flip and expect to gain any advantage out of it. On the other extreme, a sequence such as what given by digits of $\pi = 3.14159265\ldots$ may look random but is in fact perfectly structured. Indeed one can use a computer program to perfectly predict the outcomes of this sequence. A physical source, on the other hand, may have some inherent structure in it. In particular, the outcome of a physical process at a certain point might be more or less predictable, dictated by physical laws, from the outcomes observed immediately prior to that time. However, the degree of predictability may of course not be as high as in the case of π.

From a combinatorial point of view, an extractor is a combinatorial object that neutralizes any kind of structure that is inherent in a random source, and, *extracts* the "random component" out (if there is any). On the other hand, in order to be any useful, an extractor must be computationally efficient. At a first sight, it may look somewhat surprising to learn that such objects may even exist! In fact, as in the case of Ramsey graphs, the probabilistic method can be used to show that a randomly chosen function is almost surely a decent extractor. However, a random function is obviously not good enough as an extractor since the whole purpose of an extractor is to eliminate the need for pure randomness. Thus for most applications, an extractor (and more generally, condenser) is required to be efficiently computable and utilize as small amount of auxiliary pure randomness as possible.

While randomness extractors were originally studied for the main purpose of eliminating the need for pure randomness in randomized algorithms, they have found surprisingly diverse applications in different areas of combinatorics, computer science, and related fields. Among many such developments, one can mention construction of good expander graphs [161] and Ramsey graphs [9] (in fact the best known construction of Ramsey graphs can be considered a byproduct of several developments in extractor theory), communication complexity [35], Algebraic complexity theory [124], distributed computing (e.g.,

[73,128,171]), data structures (e.g., [147]), hardness of optimization problems [111,170], cryptography (see, e.g., [45]), coding theory [149], signal processing [86], and various results in structural complexity theory (e.g., [71]).

In this thesis we extend such connections to several fundamental problems related to coding theory. In the following we present a brief summary of the individual problems that are studied in each chapter.

The Wiretap Channel Problem

The wiretap channel problem studies reliable transmission of messages over a communication channel which is partially observable by a *wiretapper*. As a basic example, suppose that we wish to transmit a sensitive document over the internet. Loosely speaking, the data is transmitted in form of packets, consisting of blocks of information, through the network.

Packets may be transmitted along different paths over the network through a cloud of intermediate transmitters, called *routers*, until delivered at the destination. Now an adversary who has access to a set of the intermediate routers may be able to learn a substantial amount of information about the message being transmitted, and thereby render the communication system insecure.

A natural solution for assuring secrecy in transmission is to use a standard cryptographic scheme to encrypt the information at the source. However, the information-theoretic limitation of the adversary in the above scenario (that is, the fact that not all of the intermediate routers, but only a limited number of them are being eavesdropped) makes it possible to provably guarantee secure transmission by using a suitable encoding at the source. In particular, in a wiretap scheme, the original data is encoded at the source to a slightly redundant sequence, that is then transmitted to the recipient. As it turns out, the scheme can be designed in such a way that no information is leaked to the intruder and moreover no secrets (e.g., an encryption key) need to be shared between the two parties prior to transmission.

We study this problem in Chapter 3. The main contribution of this chapter is a construction of information-theoretically secure and optimal wiretap schemes that guarantee secrecy in various settings of the problem. In particular the scheme can be applied to point-to-point communication models as well as networks, even in presence of noise or active intrusion (i.e., when the adversary not only eavesdrops, but also alters the information being transmitted). The construction uses an explicit family of randomness extractors as the main building block.

Combinatorial Group Testing

Group testing is a classical combinatorial problem that has applications in surprisingly diverse and seemingly unrelated areas, from data structures to

coding theory to biology.

Intuitively, the problem can be described as follows: Suppose that blood tests are taken from a large population (say hundreds of thousands of people), and it is suspected that a small number (e.g., up to one thousand) carry a disease that can be diagnosed using costly blood tests. The idea is that, instead of testing blood samples one by one, it might be possible to pool them in fairly large groups, and then apply the tests on the groups without affecting reliability of the tests. Once a group is tested negative, all the samples participating in the group must be negative and this may save a large number of tests. Otherwise, a positive test reveals that at least one of the individuals in the group must be positive (though we do not learn which).

The main challenge in group testing is to design the pools in such a way to allow identification of the exact set of infected population using as few tests as possible, thereby economizing the identification process of the affected individuals. In Chapter 4 we study the group testing problem and its variations. In particular, we consider a scenario where the tests can produce highly unreliable outcomes, in which case the scheme must be designed in such a way that allows correction of errors caused by the presence of unreliable measurements. Moreover, we study a more general *threshold* variation of the problem in which a test returns positive if the number of positives participating in the test surpasses a certain threshold. This is a more reasonable model than the classical one, when the tests are not sufficiently sensitive and may be affected by dilution of the samples pooled together. In both models, we will use randomness condensers as combinatorial building blocks for construction of optimal, or nearly optimal, explicit measurement schemes that also tolerate erroneous outcomes.

Capacity Achieving Codes

The theory of error-correcting codes aims to guarantee reliable transmission of information over an unreliable communication medium, known in technical terms as a *channel*. In a classical model, messages are encoded into sequences of bits at their source, which are subsequently transmitted through the channel. Each bit being transmitted through the channel may be flipped (from 0 to 1 or vice versa) with a small probability.

Using an error-correcting code, the encoded sequence can be designed in such a way to allow correct recovery of the message at the destination with an overwhelming probability (over the randomness of the channel). However, the cost incurred by such an encoding scheme is a loss in the transmission rate, that is, the ratio between the information content of the original message and the length of the encoded sequence (or in other words, the effective number of bits transmitted per channel use).

A *capacity achieving code* is an error correcting code that essentially maximizes the transmission rate, while keeping the error probability negligible.

The maximum possible rate depends on the channel being considered, and is a quantity given by the *Shannon capacity* of the channel.

In Chapter 5, we consider a general class of communication channels (including the above example) and show how randomness condensers and extractors can be used to design capacity achieving *ensembles of* codes for them. We will then use the obtained ensembles to obtain *explicit* constructions of capacity achieving codes that allow efficient encoding and decoding as well.

Codes on the Gilbert-Varshamov Bound

While randomness extractors aim for eliminating the need for *pure* randomness in algorithms, a related class of objects known as *pseudorandom generators* aim for eliminating randomness altogether. This is made meaningful by a fundamental idea saying that randomness should be defined *relative to the observer*. The idea can be perhaps best described by an example due to Goldreich [70, Chapter 8], quoted below:

> "Alice and Bob play HEAD or TAIL in one of the following four ways. In all of them Alice flips a coin high in the air, and Bob is asked to guess its outcome *before* the coin hits the floor. The alternative ways differ by the knowledge Bob has before making his guess.
>
> In the first alternative, Bob has to announce his guess before Alice flips the coin. Clearly, in this case Bob wins with probability 1/2.
>
> In the second alternative, Bob has to announce his guess while the coin is spinning in the air. Although the outcome is *determined in principle* by the motion of the coin, Bob does not have accurate information on the motion. Thus we believe that, also in this case Bob wins with probability 1/2.
>
> The third alternative is similar to the second, except that Bob has at his disposal sophisticated equipment capable of providing accurate information on the coin's motion as well as on the environment affecting the outcome. However, Bob cannot process this information in time to improve his guess.
>
> In the fourth alternative, Bob's recording equipment is directly connected to a *powerful computer* programmed to solve the motion equations and output a prediction. It is conceivable that in such a case Bob can improve substantially his guess of the outcome of the coin."

Following the above description, in principle the outcome of a coin flip may well be deterministic. However, as long as the observer does not have enough resources to gain any advantage predicting the outcome, the coin flip should be

considered random for him. In this example, what makes the coin flip random for the observer is the inherent *hardness* (and not necessarily *impossibility*) of the prediction procedure. The theory of pseudorandom generators aim to express this line of thought in rigorous ways, and study the circumstances under which randomness can be *simulated* for a particular class of observers.

The advent of probabilistic algorithms that are unparalleled by deterministic methods, such as randomized primality testing (before the AKS algorithm [3]), polynomial identity testing and the like initially made researchers believe that the class of problems solvable by randomized polynomial-time algorithms (in symbols, BPP) might be strictly larger than those solvable in polynomial-time without the need for randomness (namely, P) and conjecture $\mathsf{P} \neq \mathsf{BPP}$. To this date, the "$\mathsf{P}$ vs. BPP" problem remains one of the most challenging problems in theoretical computer science.

Despite the initial belief, more recent research has led most theoreticians to believe otherwise, namely that $\mathsf{P} = \mathsf{BPP}$. This is supported by recent discovery of deterministic algorithms such as the AKS primality test, and more importantly, the advent of strong pseudorandom generators. In a seminal work [115], Nisan and Wigderson showed that a "hard to compute" function can be used to efficiently transform a short sequence of random bits into a much longer sequence that looks *indistinguishable* from a purely random sequence to any efficient algorithm. In short, they showed how to construct *pseudorandomness* from *hardness*. Though the underlying assumption (that certain hard functions exists) is not yet proved, it is intuitively reasonable to believe (just in the same way that, in the coin flipping game above, the hardness of gathering sufficient information for timely prediction of the outcome by Bob is reasonable to believe without proof).

In Chapter 6 we extend Nisan and Wigderson's method (originally aimed for probabilistic algorithms) to combinatorial constructions and show that, under reasonable hardness assumptions, a wide range of probabilistic combinatorial constructions can be substantially derandomized.

The specific combinatorial problem that the chapter is based on is the construction of error-correcting codes that attain the rate versus error-tolerance trade-off shown possible using the probabilistic method (namely, construction of codes on the so-called *Gilbert-Varshamov bound*). In particular, we demonstrate a small ensemble of efficiently constructible error-correcting codes almost all of which being as good as random codes (under a reasonable assumption). Even though the method is discussed for construction of error-correcting codes, it can be equally applied to numerous other probabilistic constructions; e.g., construction of optimal Ramsey graphs.

Reading Guidelines

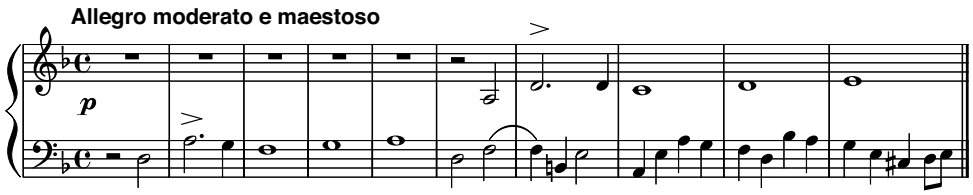
The material presented in each of the technical chapters of this thesis (Chapters 3–6) are presented independently so they can be read in any order. Since

the theory of randomness extractors plays a central role in the technical content of this thesis, Chapter 2 is devoted to an introduction to this theory, and covers some basic constructions of extractors and condenser that are used as building blocks in the main chapters. Since the extractor theory is already an extensively developed area, we will only touch upon basic topics that are necessary for understanding the thesis.

Apart from extractors, we will extensively use fundamental notions of coding theory throughout the thesis. For that matter, we have provided a brief review of such notions in Appendix A.

The additional mathematical background required for each chapter is provided when needed, to the extent of not losing focus. For a comprehensive study of the basic tools being used, we refer the reader to [5,109,112] (probability, randomness in algorithms, and probabilistic constructions), [82] (expander graphs), [8,70] (modern complexity theory), [98,103,127] (coding theory and basic algebra needed), [74] (list decoding), and [50,51] (combinatorial group testing).

Each chapter of the thesis is concluded by the opening notes of a piece of music that I truly admire.



Johann Sebastian Bach (1685–1750): The Art of Fugue BWV 1080, Contrapunctus XIV.

"Art would be useless if the world were perfect, as man wouldn't look for harmony but would simply live in it."

— *Andrei Tarkovsky*

Chapter 2

Extractor Theory

Suppose that you are given a possibly biased coin that falls heads some p fraction of times ($0 < p < 1$) and are asked to use it to "simulate" fair coin flips. A natural approach to solve this problem would be to first try to "learn" the bias p by flipping the coin a large number of times and observing the fraction of times it falls heads during the experiment, and then using this knowledge to encode the sequence of biased flips to its information-theoretic entropy.

Remarkably, back in 1951 John von Neumann [159] demonstrated a simple way to solve this problem without knowing the bias p: flip the coin twice and one of the following cases may occur:

1. The first flip shows Heads and the second Tails: output "H".
2. The first flip shows Tails and the second Heads: output "T".
3. Otherwise, repeat the experiment.

Note that the probability that the output symbol is "H" is precisely equal to it being "T", namely, $p(1-p)$. Thus, the outcome of this process represents a perfectly fair coin toss. This procedure might be somewhat wasteful; for instance, it is expected to waste half of the coin flips even if $p = 1/2$ (that is, if the coin is already fair) and that is the cost we pay for not knowing p. But nevertheless, it transforms an imperfect, not fully known, source of randomness into a perfect source of random bits.

This example, while simple, demonstrates the basic idea in what is known as "extractor theory". The basic goal in extractor theory is to improve randomness, that is, to efficiently transform a "weak" source of randomness into one with better qualities; in particular, having a higher entropy per symbol. The procedure shown above, seen as a function from the sequence of coin flips

to a Boolean function (over $\{H, T\}$) is known as an *extractor*. It is called so since it "extracts" pure randomness from a weak source.

When the distribution of the weak source is known, it is possible to use techniques from *source coding* (say Huffman or Arithmetic Coding) to compress the information to a number of bits very close to its actual entropy, without losing any of the source information. What makes extractor theory particularly challenging is the following issues:

1. An extractor knows little about the exact source distribution. Typically nothing more than a lower bound on the source entropy, and no structure is assumed on the source. In the above example, even though the source distribution was unknown, it was known to be an i.i.d. sequence (i.e., a sequence of independent, identically distributed symbols). This need not be the case in general.

2. The output of the extractor must "strongly" resemble a uniform distribution (which is the distribution with maximum possible entropy), in the sense that no statistical test (no matter how complex) should be able to distinguish between the output distribution and a purely random sequence. Note, for example, that a sequence of $n-1$ uniform and independent bits followed by the symbol "0" has $n-1$ bits of entropy, which is only slightly lower than that of n purely random bits (i.e., n). However, a simple statistical test can trivially distinguish between the two distributions by only looking at the last bit.

Since extractors and related objects (in particular, *lossless condensers*) play a central role in the technical core of this thesis, we devote this chapter to a formal treatment of extractor theory, introducing the basic ideas and some fundamental constructions. In this chapter, we will only cover basic notions and discuss a few of the results that will be used as building blocks in the rest of thesis.

2.1 Probability Distributions

2.1.1 Distributions and Distance

In this thesis we will focus on probability distributions over finite domains. Let $(\Omega, \mathsf{E}, \mathcal{X})$ be a probability space, where Ω is a finite sample space, E is the set of events (that in our work, will always consist of the set of subsets of Ω), and $\mathcal{X}$ is a probability measure. The probability assigned to each outcome $x \in \Omega$ by $\mathcal{X}$ will be denoted by $\mathcal{X}(x)$, or $\Pr_{\mathcal{X}}(x)$. Similarly, for an event $T \in \mathsf{E}$, we will denote the probability assigned to T by $\mathcal{X}(T)$, or $\Pr_{\mathcal{X}}[T]$ (when clear from the context, we may omit the subscript $\mathcal{X}$). The *support* of $\mathcal{X}$ is defined as

$$\mathsf{supp}(\mathcal{X}) := \{x \in \Omega \colon \mathcal{X}(x) > 0\}.$$

A particularly important probability measure is defined by the uniform distribution, which assigns equal probabilities to each element of Ω. We will denote the uniform distribution over Ω by $\mathcal{U}_\Omega$, and use the shorthand $\mathcal{U}_n$, for an interger $n \geq 1$, for $\mathcal{U}_{\{0,1\}^n}$. We will use the notation $X \sim \mathcal{X}$ to denote that the random variable X is drawn from the probability distribution $\mathcal{X}$.

It is often convenient to think about the probability measure as a real vector of dimension $|\Omega|$, whose entries are indexed by the elements of Ω, such that the value at the ith entry of the vector is $\mathcal{X}(i)$.

An important notion for our work is the distance between distributions. There are several notions of distance in the literature, some stronger than the others, and often the most suitable choice depends on the particular application in hand. For our applications, the most important notion is the ℓ_p distance:

Definition 2.1. Let $\mathcal{X}$ and $\mathcal{Y}$ be probability distributions on a finite domain Ω. Then for every $p \geq 1$, their ℓ_p distance, denoted by $\|\mathcal{X} - \mathcal{Y}\|_p$, is given by

$$\|\mathcal{X} - \mathcal{Y}\|_p := \left(\sum_{x \in \Omega} |\mathcal{X}(x) - \mathcal{Y}(y)|^p \right)^{1/p}.$$

We extend the distribution to the special case $p = \infty$, to denote the *point-wise* distance:

$$\|\mathcal{X} - \mathcal{Y}\|_\infty := \max_{x \in \Omega} |\mathcal{X}(x) - \mathcal{Y}(y)|.$$

The distributions $\mathcal{X}$ and $\mathcal{Y}$ are called ϵ-close with respect to the ℓ_p norm if and only if $\|\mathcal{X} - \mathcal{Y}\|_p \leq \epsilon$.

We remark that, by the Cauchy-Schwarz inequality, the following relationship between ℓ_1 and ℓ_2 distances holds:

$$\|\mathcal{X} - \mathcal{Y}\|_2 \leq \|\mathcal{X} - \mathcal{Y}\|_1 \leq \sqrt{|\Omega|} \cdot \|\mathcal{X} - \mathcal{Y}\|_2.$$

Of particular importance is the *statistical* (or *total variation*) distance. This is defined as half the ℓ_1 distance between the distributions:

$$\|\mathcal{X} - \mathcal{Y}\| := \tfrac{1}{2}\|\mathcal{X} - \mathcal{Y}\|_1.$$

We may also use the notation $\mathsf{dist}(\mathcal{X}, \mathcal{Y})$ to denote the statistical distance. We call two distributions ϵ-close if and only if their statistical distance is at most ϵ. When there is no risk of confusion, we may extend such notions as distance to the random variables they are sampled from, and, for instance, talk about two random variables being ϵ-close.

This is in a sense, a very strong notion of distance since, as the following proposition suggests, it captures the worst-case difference between the probability assigned by the two distributions to any event:

Proposition 2.2. *Let $\mathcal{X}$ and $\mathcal{Y}$ be distributions on a finite domain Ω. Then $\mathcal{X}$ and $\mathcal{Y}$ are ϵ-close if and only if for every event $T \subseteq \Omega$, $|\Pr_{\mathcal{X}}[T] - \Pr_{\mathcal{Y}}[T]| \leq \epsilon$.*

Proof. Denote by $\Omega_{\mathcal{X}}$ and $\Omega_{\mathcal{Y}}$ the following partition of Ω:

$$\Omega_{\mathcal{X}} := \{x \in \Omega \colon \mathcal{X}(x) \geq \mathcal{Y}(x)\}, \quad \Omega_{\mathcal{Y}} := \Omega \setminus T_{\mathcal{X}}.$$

Thus, $\|\mathcal{X} - \mathcal{Y}\| = 2(\Pr_{\mathcal{X}}(\Omega_{\mathcal{X}}) - \Pr_{\mathcal{Y}}(\Omega_{\mathcal{X}})) = 2(\Pr_{\mathcal{Y}}(\Omega_{\mathcal{Y}}) - \Pr_{\mathcal{X}}(\Omega_{\mathcal{Y}}))$. Let $p_1 := \Pr_{\mathcal{X}}[T \cap \Omega_{\mathcal{X}}] - \Pr_{\mathcal{Y}}[T \cap \Omega_{\mathcal{X}}]$, and $p_2 := \Pr_{\mathcal{Y}}[T \cap \Omega_{\mathcal{Y}}] - \Pr_{\mathcal{X}}[T \cap \Omega_{\mathcal{Y}}]$. Both p_1 and p_2 are positive numbers, each no more than ϵ. Therefore,

$$|\Pr_{\mathcal{X}}[T] - \Pr_{\mathcal{Y}}[T]| = |p_1 - p_2| \leq \epsilon.$$

For the reverse direction, suppose that for every event $T \subseteq \Omega$, $|\Pr_{\mathcal{X}}[T] - \Pr_{\mathcal{Y}}[T]| \leq \epsilon$. Then,

$$\|\mathcal{X} - \mathcal{Y}\|_1 = |\Pr_{\mathcal{X}}[\Omega_X] - \Pr_{\mathcal{Y}}[\Omega_X]| + |\Pr_{\mathcal{X}}[\Omega_Y] - \Pr_{\mathcal{Y}}[\Omega_Y]| \leq 2\epsilon.$$

□

An equivalent way of looking at an event $T \subseteq \Omega$ is by defining a predicate $P \colon \Omega \to \{0,1\}$ whose set of accepting inputs is T; namely, $P(x) = 1$ if and only if $x \in T$. In this view, Proposition 2.2 can be written in the following equivalent form.

Proposition 2.3. *Let $\mathcal{X}$ and $\mathcal{Y}$ be distributions on the same finite domain Ω. Then $\mathcal{X}$ and $\mathcal{Y}$ are ϵ-close if and only if, for every distinguisher $P \colon \Omega \to \{0,1\}$, we have*

$$\left| \Pr_{X \sim \mathcal{X}}[P(X) = 1] - \Pr_{Y \sim \mathcal{Y}}[P(Y) = 1] \right| \leq \epsilon.$$

□

The notion of *convex combination* of distributions is defined as follows:

Definition 2.4. Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n$ be probability distributions over a finite space Ω and $\alpha_1, \alpha_2, \ldots, \alpha_n$ be nonnegative real values that sum up to 1. Then the *convex combination*

$$\alpha_1 \mathcal{X}_1 + \alpha_2 \mathcal{X}_2 + \cdots + \alpha_n \mathcal{X}_n$$

is a distribution $\mathcal{X}$ over Ω given by the probability measure

$$\Pr_{\mathcal{X}}(x) := \sum_{i=1}^{n} \alpha_i \Pr_{\mathcal{X}_i}(x),$$

for every $x \in \Omega$.

When regarding probability distributions as vectors of probabilities (with coordinates indexed by the elements of the sample space), convex combination of distributions is merely a linear combination (specifically, a point-wise average) of their vector forms. Thus intuitively, one expects that if a probability distribution is close to a collection of distributions, it must be close to any convex combination of them as well. This is made more precise in the following proposition.

Proposition 2.5. *Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n$ be probability distributions, all defined over the same finite set Ω, that are all ϵ-close to some distribution $\mathcal{Y}$. Then any convex combination*

$$\mathcal{X} := \alpha_1 \mathcal{X}_1 + \alpha_2 \mathcal{X}_2 + \cdots + \alpha_n \mathcal{X}_n$$

is ϵ-close to $\mathcal{Y}$.

Proof. We give a proof for the case $n = 2$, which generalizes to any larger number of distributions by induction. Let $T \subseteq \Omega$ be any nonempty subset of Ω. Then we have

$$\begin{aligned} |\Pr_{\mathcal{X}}[T] - \Pr_{\mathcal{Y}}[T]| &= |\alpha_1 \Pr_{\mathcal{X}_1}[T] + (1-\alpha_1)\Pr_{\mathcal{X}_2}[T] - \Pr_{\mathcal{Y}}[T]| \\ &= |\alpha_1(\Pr_{\mathcal{Y}}[T] + \epsilon_1) + (1-\alpha_1)(\Pr_{\mathcal{Y}}[T] + \epsilon_2) - \Pr_{\mathcal{Y}}[T]|, \end{aligned}$$

where $|\epsilon_1|, |\epsilon_2| \leq \epsilon$ by the assumption that $\mathcal{X}_1$ and $\mathcal{X}_2$ are ϵ-close to $\mathcal{Y}$. Hence the distance simplifies to

$$|\alpha_1 \epsilon_1 + (1-\alpha_1)\epsilon_2|,$$

and this is at most ϵ. □

In a similar manner, it is straightforward to see that a convex combination $(1-\epsilon)\mathcal{X} + \epsilon\mathcal{Y}$ is ϵ-close to $\mathcal{X}$.

Sometimes, in order to show a claim for a probability distribution it may be easier, and yet sufficient, to write the distribution as a convex combination of "simpler" distributions and then prove the claim for the simpler components. We will examples of this technique when we analyze constructions of extractors and condensers.

2.1.2 Entropy

A central notion in the study of randomness is related to the *information content* of a probability distribution. Shannon formalized this notion in the following form:

Definition 2.6. Let $\mathcal{X}$ be a distribution on a finite domain Ω. The *Shannon entropy* of $\mathcal{X}$ (in bits) is defined as

$$H(\mathcal{X}) := \sum_{x \in \mathsf{supp}(\mathcal{X})} -\mathcal{X}(x) \log_2 \mathcal{X}(x) = \mathbb{E}_{X \sim \mathcal{X}}[-\log_2 \mathcal{X}(X)].$$

Intuitively, Shannon entropy quantifies the number of bits required to specify a sample drawn from $\mathcal{X}$ *on average*. This intuition is made more precise, for example by Huffman coding that suggest an efficient algorithm for encoding a random variable to a binary sequence whose expected length is almost equal to the Shannon entropy of the random variable's distribution (cf. [40]). For numerous applications in computer science and cryptography, however, the notion of Shannon entropy–which is an *average-case* notion–is not well suitable and a *worst-case* notion of entropy is required. Such a notion is captured by *min-entropy*, defined below.

Definition 2.7. Let $\mathcal{X}$ be a distribution on a finite domain Ω. The *min-entropy* of $\mathcal{X}$ (in bits) is defined as

$$H_\infty(\mathcal{X}) := \min_{x \in \mathsf{supp}(\mathcal{X})} -\log_2 \mathcal{X}(x).$$

Therefore, the min-entropy of a distribution is at least k if and only if the distribution assigns a probability of at most 2^{-k} to any point of the sample space (such a distribution is called a k-source). It also immediately follows by definitions that a distribution having min-entropy at least k must also have a Shannon entropy of at least k. When $\Omega = \{0,1\}^n$, we define the *entropy rate* of a distribution $\mathcal{X}$ on Ω as $H_\infty(\mathcal{X})/n$.

A particular class of probability distributions for which the notions of Shannon entropy an min-entropy coincide is *flat distributions*. A distribution on Ω is called flat if it is uniformly supported on a set $T \subseteq \Omega$; that is, if it assigns probability $1/|T|$ to all the points on T and zeros elsewhere. The Shannon- and min-entropies of such a distribution are both $\log_2 |T|$ bits.

An interesting feature of flat distributions is that their convex combinations can define any arbitrary probability distribution with a nice preservence of the min-entropy, as shown below.

Proposition 2.8. *Let K be an integer. Then any distribution $\mathcal{X}$ with min-entropy at least $\log K$ can be described as a convex combination of flat distributions with min-entropy $\log K$.*

Proof. Suppose that $\mathcal{X}$ is distributed on a finite domain Ω. Any probability distribution on Ω can be regarded as a real vector with coordinates indexed by the elements of Ω, encoding its probability measure. The set of distributions $(p_i)_{i \in \Omega}$ with min-entropy at least $\log K$ form a simplex

$$\begin{gathered}(\forall i \in \Omega)\ 0 \leq p_i \leq 1/K, \\ \textstyle\sum_{i \in \Omega} p_i = 1,\end{gathered}$$

whose corner points are flat distributions. The claim follows since every point in the simplex can be written as a convex combination of the corner points. □

2.2 Extractors and Condensers

2.2.1 Definitions

Intuitively, an extractor is a function that transforms impure randomness; i.e., a random source containing a sufficient amount of entropy, to an almost uniform distribution (with respect to a suitable distance measure; e.g., statistical distance).

Suppose that a source $\mathcal{X}$ is distributed on a sample space $\Omega := \{0,1\}^n$ with a distribution containing at least k bits of min-entropy. The goal is to construct a function $f\colon \{0,1\}^n \to \{0,1\}^m$ such that $f(\mathcal{X})$ is ϵ-close to the uniform distribution $\mathcal{U}_m$, for a negligible distance ϵ (e.g., $\epsilon = 2^{-\Omega(n)}$). Unfortunately, without having any further knowledge on $\mathcal{X}$, this task becomes impossible. To see why, consider the simplest nontrivial case where $k = n-1$ and $m = 1$, and suppose that we have come up with a function f that extracts one almost unbiased coin flip from any k-source. Observe that among the set of pre-images of 0 and 1 under f; namely, $f^{-1}(0)$ and $f^{-1}(1)$, at least one must have size 2^{n-1} or more. Let $\mathcal{X}$ be the flat source uniformly distributed on this set. The distribution $\mathcal{X}$ constructed this way has min-entropy at least $n-1$ yet $f(\mathcal{X})$ is always constant. In order to alleviate this obvious impossibility, one of the following two solutions is typically considered:

1. Assume some additional structure on the source: In the counterexample above, we constructed an opportunistic choice of the source $\mathcal{X}$ from the function f. However, in general the source obtained this way may turn out to be exceedingly complex and unstructured, and the fact that f is unable to extract any randomness from this particular choice of the source might be of little concern. A suitable way to model this observation is to require a function f that is expected to extract randomness only from a *restricted class* of randomness sources.

   The appropriate restriction in question may depend on the context for which the extractor is being used. A few examples that have been considered in the literature include:

   - Independent sources: In this case, the source $\mathcal{X}$ is restricted to be a product distribution with two or more components. In particular, one may assume the source to be the product distribution of $r \geq 2$ independent random variables $X_1, \ldots, X_r \in \{0,1\}^{n'}$ that are each sampled from an arbitrary k'-source (assuming $n = rn'$ and $k = rk'$).

- Affine sources: We assume that the source $\mathcal{X}$ is uniformly supported on an arbitrary translation of an unknown k-dimensional vector subspace of[1] $\mathbb{F}_2^n$. A further restriction of this class is known as *bit-fixing sources*. A bit-fixing source is a product distribution of n bits $(X_1, \ldots, X_n)$ where for some unknown set of coordinates positions $S \subseteq [n]$ of size k, the variables X_i for $i \in S$ are independent and uniform bits, but the rest of the X_i's are fixed to unknown binary values. In Chapter 3, we will discuss these classes of sources in more detail.
- Samplable sources: This is a class of sources first studied by Trevisan and Vadhan [153]. In broad terms, a samplable source is a source $\mathcal{X}$ such that a sample from $\mathcal{X}$ can produced out of a sequence of random and independent coin flips by a restricted computational model. For example, one may consider the class of sources of min-entropy k such that for any source $\mathcal{X}$ in the class, there is a function $f\colon \{0,1\}^r \to \{0,1\}^n$, for some $r \geq k$, that is computable by polynomial-size Boolean circuits and satisfies $f(\mathcal{U}_r) \sim \mathcal{X}$.

For restricted classes of sources such as the above examples, there are deterministic functions that are good extractors for all the sources in the family. Such deterministic functions are known as *seedless extractors* for the corresponding family of sources. For instance, an affine extractor for entropy k and error ϵ (in symbols, an affine (k, ϵ)-extractor) is a mapping $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ such that for every affine k-source $\mathcal{X}$, the distribution $f(\mathcal{X})$ is ϵ-close to the uniform distribution $\mathcal{U}_m$.

In fact, it is not hard to see that for any family of not "too many" sources, there is a function that extracts almost the entire source entropy of the sources (examples include affine k-sources, samplable k-sources, and two independent sources[2]). This can be shown by a probabilistic argument that considers a random function and shows that it achieves the desired properties with overwhelming probability.

2. Allow a short *random seed*: The second solution is to allow extractor to use a small amount of pure randomnness as a "catalyst". Namely, the extractor is allowed to require two inputs: a sample from the unknown source and a short sequence of random and independent bits that is called the *seed*. In this case, it turns out that extracting almost the entire entropy of the weak source becomes possible, without any structural assumptions on the source and using a very short independent seed. Extractors that require an auxiliary random input are called *seeded extractors*. In fact, an equivalent of looking at seeded extractors is to see

[1] Throughout the thesis, for a prime power q, we will use the notation $\mathbb{F}_q$ to denote the finite field with q elements.

[2] For this case, it suffices to count the number of independent *flat* sources.

them as *seedless* extractors that assume the source to be structured as a product distribution of two sources: an arbitrary k-source and the uniform distribution.

For the rest of this chapter, we will focus on seeded extractors. Seedless extractors (especially affine extractors) are treated in Chapter 3. A formal definition of (seeded) extractors is as follows.

Definition 2.9. A function $f\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ϵ)-*extractor* if, for every k-source $\mathcal{X}$ on $\{0,1\}^n$, the distribution $f(\mathcal{X},\mathcal{U}_d)$ is ϵ-close (in statistical distance) to the uniform distribution on $\{0,1\}^m$. The parameters n, d, k, m, and ϵ are respectively called the *input length*, *seed length*, *entropy requirement*, *output length*, and *error* of the extractor.

An important aspect of randomness extractors is their computational complexity. For most applications, extractors are required to be efficiently computable functions. We call an extractor *explicit* if it is computable in polynomial time (in its input length). Though it is rather straightforward to show existence of good extractors using probabilistic arguments, coming up with a nontrivial explicit construction can turn out a much more challenging task. We will discuss and analyze several important explicit constructions of seeded extractors in Section 2.3.

Note that, in the above definition of extractors, achieving an output length of up to d is trivial: the extractor can merely output its seed, which is guaranteed to have a uniform distribution! Ideally the output of an extractor must be "almost independent" of its seed, so that the extra randomness given in the seed can be "recycled". This idea is made precise in the notion of *strong extractors* given below.

Definition 2.10. A function $f\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a strong (k,ϵ)-*extractor* if, for every k-source $\mathcal{X}$ on $\{0,1\}^n$, and random variables $X \sim \mathcal{X}$, $Z \sim \mathcal{U}_d$, the distribution of the random variable $(X, f(X,Z))$ is ϵ-close (in statistical distance) to $\mathcal{U}_{d+m}$.

A fundamental property of strong extractors that is essential for certain applications is that, the extractor's output remains close to uniform for almost all fixings of the random seed. This is made clear by an "averaging argument" stated formally in the proposition below.

Proposition 2.11. *Consider joint distributions $\tilde{\mathcal{X}} := (\mathcal{Z},\mathcal{X})$ and $\tilde{\mathcal{Y}} := (\mathcal{Z},\mathcal{Y})$ that are ϵ-close, where $\mathcal{X}$ and $\mathcal{Y}$ are distributions on a finite domain Ω, and $\mathcal{Z}$ is uniformly distributed on $\{0,1\}^d$. For every $z \in \{0,1\}^d$, denote by $\mathcal{X}_z$ the distribution of the second coordinate of $\tilde{\mathcal{X}}$ conditioned on the first coordinate being equal to z, and similarly define $\mathcal{Y}_z$ for the distribution $\tilde{\mathcal{Y}}$. Then, for every $\delta > 0$, at least $(1-\delta)2^d$ choices of $z \in \{0,1\}^d$ must satisfy*

$$\|\mathcal{X}_z - \mathcal{Y}_z\| \leq \epsilon/\delta.$$

Proof. Clearly, for every $\omega \in \Omega$ and $z \in \{0,1\}^d$, we have $\mathcal{X}_z(\omega) = 2^d\mathcal{X}(z,\omega)$ and similarly, $\mathcal{Y}_z(\omega) = 2^d\mathcal{Y}(z,\omega)$. Moreover from the definition of statistical distance,

$$\sum_{z\in\{0,1\}^d}\sum_{\omega\in\Omega}|\mathcal{X}(z,\omega)-\mathcal{Y}(z,\omega)| \le 2\epsilon.$$

Therefore,

$$\sum_{z\in\{0,1\}^d}\sum_{\omega\in\Omega}|\mathcal{X}_z(\omega)-\mathcal{Y}_z(\omega)| \le 2^{d+1}\epsilon,$$

which can be true only if for at least $(1-\delta)$ fraction of the choices of z, we have

$$\sum_{\omega\in\Omega}|\mathcal{X}_z(\omega)-\mathcal{Y}_z(\omega)| \le 2\epsilon/\delta,$$

or in other words,

$$\|\mathcal{X}_z - \mathcal{Y}_z\| \le \epsilon/\delta.$$

This shows the claim. □

Thus, according to Proposition 2.11, for a strong (k,ϵ)-extractor

$$f\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

and a k-source $\mathcal{X}$, for $1-\sqrt{\epsilon}$ fraction of the choices of $z \in \{0,1\}^d$, the distribution $f(\mathcal{X},z)$ must be ϵ-close to uniform.

Extractors are specializations of the more general notion of *randomness condensers*. Intuitively, a condenser transforms a given weak source of randomness into a "more purified" but possibly imperfect source. In general, the output entropy of a condenser might be substantially less than the input entropy but nevertheless, the output is generally required to have a substantially higher *entropy rate*. For the extremal case of extractors, the output entropy rate is required to be 1 (since the output is required to be an almost uniform distribution). Same as extractors, condensers can be seeded or seedless, and also seeded condensers can be required to be strong (similar to strong extractors). Below we define the general notion of strong, seeded condensers.

Definition 2.12. A function $f\colon \{0,1\}^n\times\{0,1\}^d \to \{0,1\}^m$ is a *strong* $k \to_\epsilon k'$ *condenser* if for every distribution $\mathcal{X}$ on $\{0,1\}^n$ with min-entropy at least k, random variable $X \sim \mathcal{X}$ and a *seed* $Y \sim \mathcal{U}_d$, the distribution of $(Y, f(X,Y))$ is ϵ-close to a distribution $(\mathcal{U}_d, \mathcal{Z})$ with min-entropy at least $d+k'$. The parameters k, k', ϵ, $k-k'$, and $m-k'$ are called the *input entropy*, *output entropy*, *error*, the *entropy loss* and the *overhead* of the condenser, respectively. A condenser is *explicit* if it is polynomial-time computable.

Similar to strong extractors, strong condensers remain effective under almost all fixings of the seed. This follows immediately from Proposition 2.11 and is made explicit by the following corollary:

Corollary 2.13. *Let $f\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a strong $k \to_\epsilon k'$ condenser. Consider an arbitrary parameter $\delta > 0$ and a k-source $\mathcal{X}$. Then, for all but at most a δ fraction of the choices of $z \in \{0,1\}^d$, the distribution $f(\mathcal{X}, z)$ is (ϵ/δ)-close to a k'-source.* □

Typically, a condenser is only interesting if the output entropy rate k'/m is considerably larger than the input entropy rate k/n. From the above definition, an extractor is a condenser with zero overhead. Another extremal case corresponds to the case where the entropy loss of the condenser is zero. Such a condenser is called *lossless*. We will use the abbreviated term (k, ϵ)-*condenser* for a lossless condenser with input entropy k (equal to the output entropy) and error ϵ. Moreover, if a function is a (k_0, ϵ)-condenser for every $k_0 \leq k$, it is called a $(\leq k, \epsilon)$-condenser. Most known constructions of lossless condensers (and in particular, all constructions used in this thesis) are $(\leq k, \epsilon)$-condensers for their entropy requirement k.

Traditionally, lossless condensers have been used as intermediate building blocks for construction of extractors. Having a good lossless condenser available, for construction of extractors it would suffice to focus on the case where the input entropy is large. Nevertheless, lossless condensers have been proved to be useful for a variety of applications, some of which we will discuss in this thesis.

2.2.2 Almost-Injectivity of Lossless Condensers

Intuitively, an extractor is an almost "uniformly surjective" mapping. That is, the extractor mapping distributes the probability mass of the input source almost evenly among the elements of its range.

On the other hand, a lossless condenser preserves the entire source entropy on its output and intuitively, must be an almost injective function when restricted to the domain defined by the input distribution. In other words, in the mapping defined by the condenser "collisions" rarely occur and in this view, lossless condensers are useful "hashing" tools. In this section we formalize this intuition through a simple practical application.

Given a source $\mathcal{X}$ and a function f, if $f(\mathcal{X})$ has the same entropy as that of $\mathcal{X}$ (or in other words, if f is a perfectly lossless condenser for $\mathcal{X}$) we expect that from the outcome of the function, its input when sampled from $\mathcal{X}$ must be reconstructible. For flat distributions (that is, those that are uniform on their support) and considering an error for the condenser, this is shown in the following proposition. We will use this simple fact several times throughout the thesis.

Proposition 2.14. *Let $\mathcal{X}$ be a flat distribution with min-entropy $\log K$ over a finite sample space Ω and $f\colon \Omega \to \Gamma$ be a mapping to a finite set Γ.*

1. *If $f(\mathcal{X})$ is ϵ-close to having min-entropy $\log K$, then there is a set $T \subseteq \Gamma$ of size at least $(1-2\epsilon)K$ such that*

$$(\forall y \in T \text{ and } \forall x, x' \in \mathsf{supp}(\mathcal{X})) \quad f(x) = y \wedge f(x') = y \Rightarrow x = x'.$$

2. *Suppose $|\Gamma| \geq K$. If $f(\mathcal{X})$ has a support of size at least $(1-\epsilon)K$, then it is ϵ-close to having min-entropy $\log K$.*

Proof. Suppose that $\mathcal{X}$ is uniformly supported on a set $S \subseteq \Omega$ of size K, and denote by μ the distribution $f(\mathcal{X})$ over Γ. For each $y \in \Gamma$, define

$$n_y := |\{x \in \mathsf{supp}(\mathcal{X})\colon f(x) = y\}|.$$

Moreover, define $T := \{y \in \Gamma\colon n_y = 1\}$, and similarly, $T' := \{y \in \Gamma\colon n_y \geq 2\}$. Observe that for each $y \in \Gamma$ we have $\mu(y) = n_i/K$, and also $\mathsf{supp}(\mu) = T \cup T'$. Thus,

$$|T| + \sum_{y \in T'} n_y = K. \tag{2.1}$$

Now we show the first assertion. Denote by μ' a distribution on Γ with min-entropy K that is ϵ-close to μ, which is guaranteed to exist by the assumption. The fact that μ and μ' are ϵ-close implies that

$$\sum_{y \in T'} |\mu(y) - \mu'(y)| \leq \epsilon \Rightarrow \sum_{y \in T'} (n_y - 1) \leq \epsilon K.$$

In particular, this means that $|T'| \leq \epsilon K$ (since by the choice of T', for each $y \in T'$ we have $n_y \geq 2$). Furthermore,

$$\sum_{y \in T'} (n_y - 1) \leq \epsilon K \Rightarrow \sum_{y \in T'} n_y \leq \epsilon K + |T'| \leq 2\epsilon K.$$

This combined with (2.1) gives

$$|T| = K - \sum_{y \in T'} n_y \geq (1-2\epsilon)K$$

as desired.

For the second part, observe that $|T'| \leq \epsilon K$. Let μ' be any flat distribution with a support of size K that contains the support of μ. The statistical distance between μ and μ' is equal to the difference between the probability mass of the two distributions on those elements of Γ to which μ' assigns a bigger probability, namely,

$$\frac{1}{K}(\mathsf{supp}(\mu') - \mathsf{supp}(\mu)) = \frac{\sum_{y \in T'}(n_y - 1)}{K} = \frac{\sum_{y \in T'} n_y - |T'|}{K} = \frac{K - |T| - |T'|}{K},$$

where we have used (2.1) for the last equality. But $|T| + |T'| = |\mathsf{supp}(\mu)| \geq (1-\epsilon)K$, giving the required bound. ∎

As a simple application of this fact, consider the following "source coding" problem. Suppose that Alice wants to send a message x to Bob through a noiseless communication channel, and that the message is randomly sampled from a distribution $\mathcal{X}$. Shannon's source coding theorem roughly states that, there is a compression scheme that encodes x to a binary sequence y of length $H(X)$ bits on average, where $H(\cdot)$ denotes the Shannon entropy, such that Bob can perfectly reconstruct y from x (cf. [40, Chapter 5]). If the distribution $\mathcal{X}$ is known to both Alice and Bob, they can use an efficient coding scheme such as Huffman codes or Arithmetic coding to achieve this bound (up to a small constant bits of redundancy).

On the other hand, certain *universal* compression schemes are known that guarantee an optimal compression provided that $\mathcal{X}$ satisfies certain statistical properties. For instance, Lempel-Ziv coding achieves the optimum compression rate without exact knowledge of $\mathcal{X}$ provided that is defined by a stationary, ergodic process (cf. [40, Chapter 13]).

Now consider a situation where the distribution $\mathcal{X}$ is arbitrary but only known to the receiver Bob. In this case, it is known that there is no way for Alice to substantially compress her information without interaction with Bob [2]. On the other hand, if we allow interaction, Bob may simply send a description of the probability distribution $\mathcal{X}$ to Alice so she can use a classical source coding scheme to compress her information at the entropy.

Interestingly, it turns out that this task is still possible if the amount of information sent to Alice is substantially lower than what needed to fully encode the probability distribution $\mathcal{X}$. This is particularly useful if the bandwidth from Alice to Bob is substantially lower than that of the reverse direction (consider, for example, an ADSL connection) and for this reason, the problem is dubbed as the *asymmetric communication channel problem*. In particular, Watkinson et al. [160] obtain a universal scheme with $H(\mathcal{X}) + 2$ bits of communication from Alice to Bob and $n(H(\mathcal{X}) + 2)$ bits from Bob to Alice, where n is the bit-length of the message. Moreover, Adler et al. [1] obtain strong lower bounds on the number of rounds of communication between Alice and Bob.

Now let us impose a further restriction on $\mathcal{X}$ that it is uniformly supported on a set $S \subseteq \{0,1\}^n$, and Alice knows nothing about S but its size. If we disallow interaction between Alice and Bob, there would still be no deterministic way for Alice to deterministically compress her message. This is easy to observe by noting that any deterministic, and compressing, function $\varphi\colon \{0,1\}^n \to \{0,1\}^m$, where $m < n$, has an output value with as many as 2^{n-m} pre-images, and an adversarial choice of S that concentrates on the set of such pre-images would force the compression scheme to fail.

However, let us allow the encoding scheme to be randomized, and err with a small probability over the randomness of the scheme and the message. In this case, Alice can take a strong lossless condenser $f\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ for input entropy $k := \log |S|$, choose a uniformly random seed $z \in \{0,1\}^d$,

and transmit $y := (z, f(x, z))$ to Bob. Now we argue that Bob will be able to recover x from y.

Let ϵ denote the error of the condenser. Since f is a lossless condenser for $\mathcal{X}$, we know that, for $Z \sim \mathcal{U}_d$ and $X \sim \mathcal{X}$, the distribution of $(Z, f(X, Z))$ is ϵ-close to some distribution $(\mathcal{U}_d, \mathcal{Y})$, with min-entropy at least $d + k$. Thus by Corollary 2.13 it follows that, for at least $1 - \sqrt{\epsilon}$ fraction of the choices of $z \in \{0,1\}^d$, the distribution $\mathcal{Y}_z := f(\mathcal{X}, z)$ is $\sqrt{\epsilon}$-close to having min-entropy k. For any such "good seed" z, Proposition 2.14 implies that only for at most $2\sqrt{\epsilon}$ fraction of the message realizations $x \in S$ can the encoding $f(x, z)$ be confused with a different encoding $f(x', z)$ for some $x' \in S$, $x' \neq x$. Altogether we conclude that, from the encoding y, Bob can uniquely deduce x with probability at least $1 - 3\sqrt{\epsilon}$, where the probability is taken over the randomness of the seed and the message distribution $\mathcal{X}$.

The amount of communication in this encoding scheme is $m+d$ bits. Using an optimal lossless condenser for f, the encoding length becomes $k + O(\log n)$ with a polynomially small (in n) error probability (where the exponent of the polynomial is arbitrary and affects the constant in the logarithmic term). On the other hand, with the same error probability, the explict condenser of Theorem 4.19 would give an encoding length $k + O(\log^3 n)$. Moreover, the explicit condenser of Theorem 2.22 results in length $k(1 + \alpha) + O_\alpha(\log n)$ for any arbitrary constant $\alpha > 0$.

2.3 Constructions

We now turn to explicit constructions of strong extractors and lossless condensers.

Using probabilistic arguments, Radhakrishan and Ta-Shma [122] showed that, for every k, n, ϵ, there is a strong (k, ϵ)-extractor with seed length $d = \log(n - k) + 2\log(1/\epsilon) + O(1)$ and output length $m = k - 2\log(1/\epsilon) - O(1)$. In particular, a random function achieves these parameters with probability $1 - o(1)$. Moreover, their result show that this trade-off is almost the best one can hope for.

Similar trade-offs are known for lossless condensers as well. Specifically, the probabilistic construction of Radhakrishan and Ta-Shma has been extended to the case of lossless condensers by Capalbo et al. [23], where they show that a random function is with high probability a strong lossless (k, ϵ)-condenser with seed length $d = \log n + \log(1/\epsilon) + O(1)$ and output length $m = k + \log(1/\epsilon) + O(1)$. Moreover, this tradeoff is almost optimal as well.

In this section, we introduce some important *explicit* constructions of both extractors and lossless condensers that are used as building blocks of various constructions in the thesis. In particular, we will discuss extractors and lossless condensers obtained by the Leftover Hash Lemma, Trevisan's extractor, and a lossless condenser due to Guruswami, Umans, and Vadhan.

2.3.1 The Leftover Hash Lemma

One of the foremost explicit constructions of extractors is given by the *Leftover Hash Lemma* first stated by Impagliazzo, Levin, and Luby [84]. This extractor achieves an optimal output length $m = k - 2\log(1/\epsilon)$ albeit with a substantially large seed length $d = n$. Moreover, the extractor is a linear function for every fixing of the seed. In its general form, the lemma states that any *universal family of hash functions* can be transformed into an explicit extractor. The universality property required by the hash functions is captured by the following definition.

Definition 2.15. A family of functions $\mathcal{H} = \{h_1, \ldots, h_D\}$ where $h_i \colon \{0,1\}^n \to \{0,1\}^m$ for $i = 1, \ldots, D$ is called *universal* if, for every fixed choice of $x, x' \in \{0,1\}^n$ such that $x \neq x'$ and a uniformly random $i \in [D] := \{1, \ldots, D\}$ we have

$$\Pr_i[h_i(x) = h_i(x')] \leq 2^{-m}.$$

One of the basic examples of universal hash families is what we call *the linear family*, defined as follows. Consider an arbitrary isomorphism $\varphi \colon \mathbb{F}_2^n \to \mathbb{F}_{2^n}$ between the vector space $\mathbb{F}_2^n$ and the extension field $\mathbb{F}_{2^n}$, and let $0 < m \leq n$ be an arbitrary integer. The linear family $\mathcal{H}_{\mathsf{lin}}$ is the set $\{h_\alpha \colon \alpha \in \mathbb{F}_{2^n}\}$ of size 2^n that contains a function for each element of the extension field $\mathbb{F}_{2^n}$. For each α, the mapping h_α is given by

$$h_\alpha(x) := (y_1, \ldots, y_m), \text{ where } (y_1, \ldots, y_n) := \varphi^{-1}(\alpha \cdot \varphi(x)).$$

Observe that each function h_α can be expressed as a linear mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. Below we show that this family is pairwise independent.

Proposition 2.16. *The linear family $\mathcal{H}_{\mathsf{lin}}$ defined above is universal.*

Proof. Let x, x' be different elements of $\mathbb{F}_{2^n}$. Consider the mapping $f \colon \mathbb{F}_{2^n} \to \mathbb{F}_2^m$ defined as

$$f(x) := (y_1, \ldots, y_m), \text{ where } (y_1, \ldots, y_n) := \varphi^{-1}(x),$$

which truncates the binary representation of a field element from $\mathbb{F}_{2^n}$ to m bits. The probability we are trying to estimate in Definition 2.15 is, for a uniformly random $\alpha \in \mathbb{F}_{2^n}$,

$$\Pr_{\alpha \in \mathbb{F}_{2^n}}[f(\alpha \cdot x) = f(\alpha \cdot x')] = \Pr_{\alpha \in \mathbb{F}_{2^n}}[f(\alpha \cdot (x - x')) = 0].$$

But note that $x - x'$ is a nonzero element of $\mathbb{F}_{2^n}$, and thus, for a uniformly random α, the random variable αx is uniformly distributed on $\mathbb{F}_{2^n}$. It follows that

$$\Pr_{\alpha \in \mathbb{F}_{2^n}}[f(\alpha \cdot (x - x')) = 0] = 2^{-m},$$

implying that $\mathcal{H}_{\mathsf{lin}}$ is a universal family. □

Now we are ready to state and prove the Leftover Hash Lemma. We prove a straightforward generalization of the lemma which shows that universal hash families can be used to construct not only strong extractors, but also lossless condensers.

Theorem 2.17. *(Leftover Hash Lemma) Let $\mathcal{H} = \{h_i \colon \mathbb{F}_2^n \to \mathbb{F}_2^m\}_{i \in \mathbb{F}_2^d}$ be a universal family of hash functions with 2^d elements indexed by binary vectors of length d, and define the function $f \colon \mathbb{F}_2^n \times \mathbb{F}_2^d \to \mathbb{F}_2^m$ as $f(x,z) := h_z(x)$. Then*

1. *For every k, ϵ such that $m \leq k - 2\log(1/\epsilon)$, the function f is a strong (k,ϵ)-extractor, and*
2. *For every k, ϵ such that $m \geq k + 2\log(1/\epsilon)$, the function f is a strong lossless (k,ϵ)-condenser.*

In particular, by choosing $\mathcal{H} = \mathcal{H}_{\mathsf{lin}}$, it is possible to get explicit extractors and lossless condensers with seed length $d = n$.

Proof. Considering Proposition 2.8, it suffices to show the claim when $\mathcal{X}$ is a flat distribution on a support of size $K := 2^k$. Define $M := 2^m$, $D := 2^d$, and let μ be any flat distribution over $\mathbb{F}_2^{d+m}$ such that $\mathsf{supp}(\mathcal{X}) \subseteq \mathsf{supp}(\mu)$, and denote by $\mathcal{Y}$ the distribution of $(Z, f(X,Z))$ over $\mathbb{F}_2^{d+m}$ where $X \sim \mathcal{X}$ and $Z \sim \mathcal{U}_d$. We will first upper bound the ℓ_2 distance of the two distributions $\mathcal{Y}$ and μ, that can be expressed as follows:

$$\begin{aligned}
\|\mathcal{Y} - \mu\|_2^2 &= \sum_{x \in \mathbb{F}_2^{d+m}} (\mathcal{Y}(x) - \mu(x))^2 \\
&= \sum_x \mathcal{Y}(x)^2 + \sum_x \mu(x)^2 - 2\sum_x \mathcal{Y}(x)\mu(x) \\
&\overset{(a)}{=} \sum_x \mathcal{Y}(x)^2 + \frac{1}{|\mathsf{supp}(\mu)|} - \frac{2}{|\mathsf{supp}(\mu)|}\sum_x \mathcal{Y}(x) \\
&= \sum_x \mathcal{Y}(x)^2 - \frac{1}{|\mathsf{supp}(\mu)|}, \qquad (2.2)
\end{aligned}$$

where (a) uses the fact that μ assigns probability $1/|\mathsf{supp}(\mu)|$ to exactly $|\mathsf{supp}(\mu)|$ elements of $\mathbb{F}_2^{d+m}$ and zeros elsewhere.

Now observe that $\mathcal{Y}(x)^2$ is the probability that two independent samples drawn from $\mathcal{Y}$ turn out to be equal to x, and thus, $\sum_x \mathcal{Y}(x)^2$ is the *collision probability* of two independent samples from $\mathcal{Y}$, which can be written as

$$\sum_x \mathcal{Y}(x)^2 = \Pr_{Z,Z',X,X'}[(Z, f(X,Z)) = (Z', f(X',Z'))],$$

where $Z, Z' \sim \mathbb{F}_2^d$ and $X, X' \sim \mathcal{X}$ are independent random variables. We can rewrite the collision probability as

$$\begin{aligned}
\sum_x \mathcal{Y}(x)^2 &= \Pr[Z = Z'] \cdot \Pr[f(X,Z) = f(X',Z') \mid Z = Z'] \\
&= \frac{1}{D} \cdot \Pr_{Z,X,X'}[h_Z(X) = h_Z(X')] \\
&= \frac{1}{D} \cdot (\Pr[X = X'] + \frac{1}{K^2} \sum_{\substack{x,x' \in \mathsf{supp}(\mathcal{X}) \\ x \neq x'}} \Pr_Z[h_Z(x) = h_Z(x')]) \\
&\overset{(b)}{\leq} \frac{1}{D} \cdot (\frac{1}{K} + \frac{1}{K^2} \sum_{\substack{x,x' \in \mathsf{supp}(\mathcal{X}) \\ x \neq x'}} \frac{1}{M}) \leq \frac{1}{DM} \cdot (1 + \frac{M}{K}),
\end{aligned}$$

where (b) uses the assumption that $\mathcal{H}$ is a universal hash family. Plugging the bound in (2.2) implies that

$$\|\mathcal{Y} - \mu\|_2 \leq \frac{1}{\sqrt{DM}} \cdot \sqrt{1 - \frac{DM}{|\mathsf{supp}(\mu)|} + \frac{M}{K}}.$$

Observe that both $\mathcal{Y}$ and μ assign zero probabilities to elements of $\{0,1\}^{d+m}$ outside the support of μ. Thus using Cauchy-Schwarz on a domain of size $\mathsf{supp}(\mu)$, the above bound implies that the statistical distance between $\mathcal{Y}$ and μ is at most

$$\frac{1}{2} \cdot \sqrt{\frac{|\mathsf{supp}(\mu)|}{DM}} \cdot \sqrt{1 - \frac{DM}{|\mathsf{supp}(\mu)|} + \frac{M}{K}}. \tag{2.3}$$

Now, for the first part of the theorem, we specialize μ to the uniform distribution on $\{0,1\}^{d+m}$, which has a support of size DM, and note that by the assumption that $m \leq k - 2\log(1/\epsilon)$ we will have $M \leq \epsilon^2 K$. Using (2.3), it follows that $\mathcal{Y}$ and μ are $(\epsilon/2)$-close.

On the other hand, for the second part of the theorem, we specialize μ to any flat distribution on a support of size DK containing $\mathsf{supp}(\mathcal{Y})$ (note that, since $\mathcal{X}$ is assumed to be a flat distribution, $\mathcal{Y}$ must have a support of size at most DK). Since $m \geq k + 2\log(1/\epsilon)$, we have $K = \epsilon^2 M$, and again (2.3) implies that $\mathcal{Y}$ and μ are $(\epsilon/2)$-close. □

2.3.2 Trevisan's Extractor

One of the most important explicit constructions of extractors is due to Trevisan [152]. Since we will use this extractor at several points in the thesis, we dedicate this section to sketch the main ideas behind this important construction.

Trevisan's extractor can be thought of as an "information-theoretic" variation of Nisan-Wigderson's pseudorandom generator that will be discussed in detail in Chapter 6. For the purpose of this exposition, we will informally demonstrate how Nisan-Wigderson's generator works and then discuss Trevisan's extractor from a coding-theoretic perspective.

Loosely speaking, a pseudorandom generator is an efficient and deterministic function (where the exact meaning of "efficient" may vary depending on the context) that transforms a statistically uniform distribution on d bits to a distribution on m bits, for some $m \gg d$, that "looks random" to any "restricted" distinguisher. Again the precise meaning of "looking random" and the exact restriction of the distinguisher may vary. In particular, we require the output distribution $\mathcal{X}$ of the pseudorandom generator to be such that, for every restricted distinguisher $D\colon \{0,1\}^m \to \{0,1\}$, we have

$$\left| \Pr_{X\sim\mathcal{X}}[D(X)=1] - \Pr_{Y\sim\mathcal{U}_m}[D(Y)=1] \right| \leq \epsilon,$$

where ϵ is a negligible bias. Recall that, in in light of Proposition 2.3, this is very close to what we expect from the output distribution of an extractor, except that for the case of pseudorandom generators the distinguisher D cannot be an arbitrary function. Indeed, when $m > d$, the output distribution of a pseudorandom generator cannot be close to uniform and is always distinguishable by *some* distinguisher. The main challenge in construction of a pseudorandom gnerator is to exclude the possibility of such a distinguisher to be included in the restricted class of functions into consideration. As a concrete example, one may require a pseudorandom generator to be a polynomial-time computable function whose output is a sequence of length d^2 that is indistinguishable by linear-sized Boolean circuits with a bias better than d^{-2}.

Nisan and Wigderson observed that the hardness of distinguishing the output distribution from uniform can be derived from a hardness assumption that is inherent in the way the pseudorandom generator itself is computed. In a way, their construction shows how to "trade" computational hardness with pseudorandomness. In a simplified manner, a special instantiation of this generator can be described as follows: Suppose that a Boolean predicate $f\colon \{0,1\}^d \to \{0,1\}$ is hard to compute on average by "small" Boolean circuits; meaning that no circuit consisting of a sufficiently small number of gates (as determined by a *security parameter*) is able to compute f substantially better than a trivial circuit that always outputs a constant value. Then, given a random seed $Z \in \{0,1\}^d$, the sequence $(Z, f(Z))$ is pseudorandom for small circuits. The reason can be seen by contradiction. Let us suppose that for some distinguisher D, we have

$$\left| \Pr_{X\sim\mathcal{X}}[D(X)=1] - \Pr_{Y\sim\mathcal{U}_m}[D(Y)=1] \right| > \epsilon.$$

By the following simple proposition, such a distinguisher can be transformed into a predictor for the hard function f.

Proposition 2.18. *Consider predicates $f\colon \mathbb{F}_2^d \to \mathbb{F}_2$ and $D\colon \mathbb{F}_2^{d+1} \to \mathbb{F}_2$ and suppose that*

$$\left| \Pr_{X \sim \mathcal{U}_d}[D(X, f(X)) = 1] - \Pr_{Y \sim \mathcal{U}_{d+1}}[D(Y) = 1] \right| > \epsilon.$$

Then, there are fixed choices of $a_0, a_1 \in \mathbb{F}_2$ such that

$$\Pr_{X \sim \mathcal{U}_d}[D(X, a_0) + a_1 = f(X)] > \frac{1}{2} + \epsilon.$$

Proof. Without loss of generality, assume that the quantity inside the absolute value is non-negative (otherwise, one can reason about the negation of D). Consider the following randomized algorithm A that, given $x \in \mathbb{F}_2^d$, tries to predict $f(X)$: Flip a random coin $r \in F_2$. If $r = 1$, output r and otherwise output $\bar{r}$.

Intuitively, the algorithm A tries to make a random guess for $f(X)$, and then feeds it to the distinguisher. As D is more likely to output 1 when the correct value of $f(X)$ is supplied, A takes the acceptance of x as an evidence that the random guess r has been correct (and vice versa). The precise analysis can be however done as follows.

$$\begin{aligned}
\Pr_{X,r}[A(X) = f(X)] &= \frac{1}{2} \Pr_{X,r}[A(X) = f(X) \mid r = f(X)] + \\
&\quad \frac{1}{2} \Pr_{X,r}[A(X) = f(X) \mid r \neq f(X)] \\
&= \frac{1}{2} \Pr_{X,r}[D(X, r) = 1 \mid r = f(X)] + \\
&\quad \frac{1}{2} \Pr_{X,r}[D(X, r) = 0 \mid r \neq f(X)] \\
&= \frac{1}{2} \Pr_{X,r}[D(X, r) = 1 \mid r = f(X)] + \\
&\quad \frac{1}{2}(1 - \Pr_{X,r}[D(X, r) = 1 \mid r \neq f(X)]) \\
&= \frac{1}{2} + \Pr_{X,r}[D(X, r) = 1 \mid r = f(X)] - \\
&\quad \frac{1}{2} \left(\Pr_{X,r}[D(X, r) = 1 \mid r = f(X)] + \right. \\
&\quad \left. \Pr_{X,r}[D(X, r) = 1 \mid r \neq f(X)] \right) \\
&= \frac{1}{2} + \Pr_{X}[D(X, f(X)) = 1] - \Pr_{X,r}[D(X, r) = 1]
\end{aligned}$$

$$> \frac{1}{2} + \epsilon.$$

Therefore, by averaging, for some fixed choice of r the probability must remain above $\frac{1}{2} + \epsilon$, implying that one of the functions $D(X, 0)$, $D(X, 1)$ or their negations must be as good a predictor for $f(X)$ as A is. □

Since the complexity of the predictor is about the same as that of the distinguisher D, and by assumption f cannot be computed by small circuits, we conclude that the outcome of the generator must be indistinguishable from uniform by small circuits. Nisan and Wigderson generalized this idea to obtain generators that output a long sequence of bits that is indistinguishable from having a uniform distribution. In order to obtain more than one pseudorandom bit from the random seed, they evaluate the hard function f on carefully chosen subsequences of the seed (for this to work, the input length of f is assumed to be substantially smaller than the seed length d).

An important observation in Trevisan's work is that Nisan-Wigderson's pseudorandom generator is a *black-box* construction. Namely, the generator merely computes the hard function f at suitably chosen points without caring much about how this computation is implemented. Similarly, the analysis uses the distinguisher D as a black-box. If f is computable in polynomial time, then so is the generator (assuming that it outputs polynomially many bits), and if f is hard against small circuits, the class of circuits of about the same size must be fooled by the generator.

How can we obtain an extractor from Nisan-Wigderson's construction? Recall that the output distribution of an extractor must be indistinguishable from uniform by *all* circuits, and not only small ones. Adapting Nisan-Wigderson's generator for this requirement means that we will need a function f that is hard for all circuits, something which is obviously impossible. However, this problem can be resolved if we take *many* hard functions instead of one, and enforce the predictor to simultaneously predict all functions with a reasonable bias. More precisely, statistical indistinguishability can be obtained if the function f is sampled from a random distribution, and that is exactly how Trevisan's extractor uses the supplied weak source. In particular, the extractor regards the sequence obtained from the weak source as the truth table of a randomly chosen function, and then applies Nisan-Wigderson's construction relative to that function.

The exact description of the extractor is given in Construction 2.1. The extractor assumes the existence of a suitable list-decodable code (see Appendix A for the terminology) as well as a *combinatorial design*. Intuitively, a combinatorial design is a collection of subsets of a universe such that their pairwise intersections are small. We will study designs more closely in Chapter 4. In order to obtain a polynomial-time computable extractor, we need an

- *Given:* A random sample $X \sim \mathcal{X}$, where $\mathcal{X}$ is a distribution on $\{0,1\}^n$ with min-entropy at least k, and a uniformly distributed random seed $Z \sim \mathcal{U}_d$ of length d. Moreover, the extractor assumes a $(\frac{1}{2} - \delta, \ell)$ list-decodable binary code $\mathcal{C}$ of length N (a power of two) and size 2^n, and a combinatorial design $\mathcal{S} := \{S_1, \ldots, S_m\}$, where
  - For all $i \in [m]$, $S_i \subseteq [d]$, $|S_i| = \log_2 N$, and
  - For all $1 \leq i < j \leq m$, $|S_i \cap S_j| \leq r$.
- *Output:* A binary string $E(X, Z)$ of length m.
- *Construction:* Denote the encoding of X under $\mathcal{C}$ by $C(X)$. For each $i \in [m]$, the subsequence of Z picked by the coordinate positions in S_i (denoted by $Z|_i$) is a string of length $\log_2 N$ and can be regarded as an integer in $[N]$. Let $C_i(X)$ denote the bit at the $(Z|_i)$th position of the encoding $C(X)$. Then,

$$E(X, Z) := (C_1(X), \ldots, C_m(X)).$$

Construction 2.1: Trevisan's extractor $E\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$.

efficient construction of the underlying list-decodable code and combinatorial design.

An analysis of Trevisan's construction is given by the following theorem, which is based on the original analysis of [152].

Theorem 2.19. *Trevisan's extractor (as described in Construction 2.1) is a strong (k, ϵ)-extractor provided that $\epsilon \geq 2m\delta$ and $k > d + m2^{r+1} + \log(\ell/\epsilon) + 3$.*

Proof. In light of Proposition 2.8, it suffices to show the claim when $\mathcal{X}$ is a flat distribution. Suppose for the sake of contradiction that the distribution of $(Z, E(X, Z))$ is not ϵ-close to uniform. Without loss of generality, and using Proposition 2.3, this means that there is a distinguisher $D\colon \{0,1\}^m \to \{0,1\}$ such that

$$\Pr_{X,Z}[D(Z, E(X, Z)) = 1] - \Pr_{Z,U \sim \mathcal{U}_m}[D(Z, U) = 1] > \epsilon, \tag{2.4}$$

where $U = (U_1, \ldots, U_m)$ is a sequence of uniform and independent random bits. Let $X' \subseteq \mathsf{supp}(\mathcal{X})$ denote the set of inputs on the support of $\mathcal{X}$ that satisfy

$$\Pr_{Z}[D(Z, E(x, Z)) = 1] - \Pr_{Z,U}[D(Z, U) = 1] > \frac{\epsilon}{2}, \tag{2.5}$$

Observe that the size of X' must be at least $\frac{\epsilon}{2}|\mathsf{supp}(\mathcal{X})| = \epsilon 2^{k-1}$, since otherwise (2.4) cannot be satisfied. In the sequel, fix any $x \in X'$.

For $i = 0, \ldots, m$, define a *hybrid sequence* H_i as the random variable $H_i := (Z, C_1(x), \ldots, C_i(x), U_{i+1}, \ldots, U_m)$. Thus, H_0 is a uniformly random bit sequence and H_m has the same distribution as $(Z, E(x, Z))$. For $i \in [m]$, define

$$\delta_i := \Pr[D(H_i) = 1] - \Pr[D(H_{i-1}) = 1],$$

where the probability is taken over the randomness of Z and U. Now we can rewrite (2.5) as

$$\Pr[D(H_m) = 1] - \Pr[D(H_0) = 1] > \frac{\epsilon}{2},$$

or equivalently,

$$\sum_{i=1}^{m} \delta_i > \frac{\epsilon}{2}.$$

Therefore, for some $i \in [m]$, we must have $\delta_i > \epsilon/(2m) =: \epsilon'$. Fix such an i, and recall that we have

$$\begin{aligned}(2.6)\quad \Pr[D(Z, C_1(x), \ldots, C_i(x), U_{i+1}, \ldots, U_m) = 1] - \\ \Pr[D(Z, C_1(x), \ldots, C_{i-1}(x), U_i, \ldots, U_m) = 1] > \epsilon'.\end{aligned}$$

Now observe that there is a fixing $U_{i+1} = u_{i+1}, \ldots, U_m = u_m$ of the random bits $U_{i+1}, \ldots, U_m$ that preserves the above bias. In a similar way as we defined the subsequence $Z|_i$, denote by $Z|_{\bar{i}}$ the subsequence of Z obtained by removing the coordinate positions of Z picked by S_i. Now we note that $C_i(x)$ depends only on x and $Z|_i$ and is in particular independent of $Z|_{\bar{i}}$. Furthermore, one can fix $Z|_{\bar{i}}$ (namely, the portion of the random seed outside S_i) such that the bias in (2.6) is preserved. In other words, there is a string $z' \in \{0,1\}^{d-|S_i|}$ such that

$$\begin{aligned}\Pr[D(Z, C_1(x), \ldots, C_i(x), u_{i+1}, \ldots, u_m) = 1 \mid (Z|_{\bar{i}}) = z'] - \\ \Pr[D(Z, C_1(x), \ldots, C_{i-1}(x), U_i, u_{i+1}, \ldots, u_m) = 1 \mid (Z|_{\bar{i}}) = z'] > \epsilon',\end{aligned}$$

where the randomness is now only over U_i and $Z|_i$, and all other random variables are fixed to their appropriate values. Now, Proposition 2.18 can be used to show that, under the above fixings, there is a fixed choice of bits $a_0, a_1 \in \mathbb{F}_2$ such that D can be transformed into a predictor for $C_i(x)$; namely, so that

$$\Pr_Z[D(Z, C_1(x), \ldots, C_{i-1}(x), a_0, u_{i+1}, \ldots, u_m) + a_1 = C_i(x) \mid (Z|_{\bar{i}}) = z'] > \frac{1}{2} + \epsilon'.$$

Since $Z|_i$ is a uniformly distributed random variable, the above probability can be interpreted in coding-theoretic ways as follows: By running through all the

N possibilities of $Z|_i$, the predictor constructed from D can correctly recover the encoding $C(x)$ at more than $\frac{1}{2}+\epsilon'$ fraction of the positions. Therefore, the distinguisher D can be transformed into a word $w \in \mathbb{F}_2^N$ that has an agreement above $\frac{1}{2}+\frac{\epsilon}{2m}$ with $C(x)$.

Now a crucial observation is that the word w can be obtained from D without any knowledge of x, as long a correct "advice" string consisting of the appropriate fixings of $i, u_{i+1}, \ldots, u_m, a_0, a_1, z'$, and the truth tables of $C_1(x), \ldots, C_{i-1}(x)$ as functions of $Z|_i$ are available. Here is where the small intersection property of the design $\mathcal{S}$ comes to play: Each $C_j(x)$ (when $j \neq i$) depends on at most r of the bits in $Z|_i$, and therefore, $C_j(x)$ as a function of $Z|_i$ can be fully described by its evaluation on at most 2^r points (that can be much smaller than $2^{|S_i|} = N$). This means that the number of possibilities for the advice string is at most

$$m \cdot 2^m \cdot 4 \cdot 2^{d-\log N} \cdot 2^{m2^r} = \frac{m}{N} \cdot 2^{d+m(2^r+1)+2} \leq 2^{d+m(2^{r+1})+2} =: T.$$

Therefore, regardless of the choice of $x \in X'$, there are words $w_1, \ldots, w_T \in \mathbb{F}_2^N$ (one for each possibility of the advice string) such that at least one (corresponding to the "correct" advice) has an agreement better than $\frac{1}{2}+\epsilon'$ with $C(x)$. This, in turn, implies that there is a set $X'' \subseteq X'$ of size at least $|X'|/T \geq \epsilon 2^{k-1}/T$ and a fixed $j \in [T]$ such that, for every $x \in X''$, the codeword $C(x)$ has an agreement better than $\frac{1}{2}+\epsilon'$ with w_j. As long as $\delta \leq \epsilon'$, the number of such codewords can be at most ℓ (by the list-decodability of $\mathcal{C}$), and we will reach to the desired contradiction (completing the proof) if the list size ℓ is small enough; specifically, if

$$\ell < \frac{\epsilon 2^{k-1}}{T},$$

which holds by the assumption of the theorem. □

By an appropriate choice of the underlying combinatorial design $\mathcal{S}$ and the list-decodable code $\mathcal{C}$ (namely, concatenation of the Reed-Solomon code and the Hadamard code as described in Section A.5), Trevisan [152] obtained a strong extractor with output length $k^{1-\alpha}$, for any fixed constant $\alpha > 0$, and seed length $d = O(\log^2(n/\epsilon)/\log_k)$. In a subsequent work, Raz, Reingold and Vadhan observed that a weaker notion of combinatorial designs suffice for this construction to work. Using this idea and a careful choice of the list-decodable code $\mathcal{C}$, they managed to improve Trevisan's extractor so that it extracts almost the entire source entropy. Specifically, their imrpovement can be summarized as follows.

Theorem 2.20. [123] *For every $n, k, m \in \mathbb{N}$, $(m \leq k \leq n)$ and $\epsilon > 0$, there is an explicit strong (k, ϵ)-extractor $\mathsf{Tre}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log^2(n/\epsilon) \cdot \log(1/\alpha))$, where $\alpha := k/(m-1) - 1$ must be less than $1/2$.* □

- *Given:* A random sample $X \sim \mathcal{X}$, where $\mathcal{X}$ is a distribution on $\mathbb{F}_q^n$ with min-entropy at least k, and a uniformly distributed random seed $Z \sim \mathcal{U}_{\mathbb{F}_q}$ over $\mathbb{F}_q$.
- *Output:* A vector $C(X, Z)$ of length ℓ over $\mathbb{F}_q$.
- *Construction:* Take any irreducible univariate polynomial g of degree n over $\mathbb{F}_q$, and interpret the input X as the coefficient vector of a random univariate polynomial F of degree $n-1$ over $\mathbb{F}_q$. Then, for an integer parameter h, the output is given by

$$C(X, Z) := (F(Z), F_1(Z), \ldots, F_{\ell-1}(Z)),$$

where we have used the shorthand $F_i := F^{h^i} \mod g$.

Construction 2.2: Guruswami-Umans-Vadhan's Condenser $C\colon \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^m$.

Observe that, as long as the list-decodable code $\mathcal{C}$ is linear, Trevisan's extractor (as well as its improvement above) becomes linear as well, meaning that it can be described as a linear function of the weak source for every fixed choice of the seed. We will make crucial use of this observation at several points in the thesis.

2.3.3 Guruswami-Umans-Vadhan's Condenser

One of the important constructions of lossless condensers that we will use in this thesis is the coding-theoretic construction of Guruswami, Umans and Vadhan [78]. In this section, we discuss the construction (Construction 2.2) and its analysis (Theorem 2.22).

We remark that this construction is inspired by a variation of Reed-Solomon codes due to Parvaresh and Vardy [118]. Specifically, for a given $x \in \mathbb{F}_q^n$, arranging the outcomes of the condenser $C(x, z)$ for all possibilities of the seed $z \in \mathbb{F}_q$ results in the encoding of the input x using a Parvaresh-Vardy code. Moreover, Parvaresh-Vardy codes are equipped with an efficient list-decoding algorithm that is implicit in the analysis of the condenser. The main technical part of the analysis is given by the following theorem.

Theorem 2.21. [78] *The mapping defined in Construction 2.2 is a strong (k, ϵ) lossless condenser with error $\epsilon := (n-1)(h-1)\ell/q$, provided that $\ell \geq k/\log h$ (thus, under the above conditions the mapping becomes a strong $(\leq k, \epsilon)$-condenser as well).*

Proof. Without loss of generality (using Proposition 2.8), assume that $\mathcal{X}$ is uniformly distributed on a subset of $\mathbb{F}_q^n$ of size $K := 2^k$. Let $D := q - (n-1)(h-1)\ell$. Define the random variable

$$Y := (Z, F(Z), F_1(Z), \ldots, F_{\ell-1}(Z)),$$

and denote by $T \subseteq \mathbb{F}_q^{\ell+1}$ the set that supports the distribution of Y; i.e., the set of vectors in $\mathbb{F}_q^{\ell+1}$ for which Y has a nonzero probability of being assigned to. Our goal is to show that $|T| \geq DK$. Combined with the second part of Proposition 2.14, this will prove the theorem, since we will know that the distribution of $(Z, C(X,Z))$ has a support of size at least $(1-\epsilon)q2^k$.

Assume, for the sake of contradiction, that $|T| < DK$. Then the set of points in T can be interpolated by a nonzero multivaraite low-degree polynomial of the form

$$Q(z, z_1, \ldots, z_\ell) = \sum_{i=0}^{D-1} z^i Q_i'(z_1, \ldots, z_\ell),$$

where each monomial $z_1^{j_1} \cdots z_\ell^{j_\ell}$ in every Q_i' has weighted degree $j_1 + hj_2 + h^2 j_3 + \cdots + h^{\ell-1} j_\ell$ at most $K - 1 < h^\ell$ and individual degrees less than h (this condition can be assured by taking $j_1, \ldots, j_\ell$ to be the integer representation of an integer between 0 and $K-1$). Note that Q can be described by its DK coefficients, and each point on T specifies a linear constraint on their choice. Since the number of constraints is less than the number of unknowns, we know that a nonzero polynomial Q vanishes on the set T. Fix a nonzero choice of Q that has the lowest degree in the first variable z. This assures that if we write down Q as

$$Q(z, z_1, \ldots, z_\ell) = \sum_{j=(j_1,\ldots,j_\ell)} Q_j(z) z_1^{j_1} \cdots z_\ell^{j_\ell},$$

the polynomials $Q_j(z)$ do not have common irreducible factors (otherwise we could divide by the common factor and contradict minimality of the degree). In particular at least one of the Q_j's must be nonzero modulo the irreducible polynomial g.

Now consider the set S of univariate polynomials of degree less than n chosen so that

$$f \in S \Leftrightarrow (\forall z \in \mathbb{F}_q) \colon (z, f(z), f_1(z), \ldots, f_{\ell-1}(z)) \in T,$$

where, similarly as before, we have used the shorthand f_i for $(f^{h^i} \bmod g)$. Note that, if we regard $\mathsf{supp}(\mathcal{X})$ as a set of low-degree univariate polynomials, by construction of the condenser this set must be contained in S. Therefore, to reach the desired contradiction, it suffices to show that $|S| < K$.

Let f be any polynomial in S. By the definition of S, the univariate polynomial $Q(z, f(z), f_1(z), \ldots, f_{\ell-1}(z))$ must have q zeros (namely, all the

elements of $\mathbb{F}_q$). But the total degree of this polynomial is at most $D - 1 + (n-1)(h-1)\ell = q - 1$, and thus, the polynomial must be identically zero, and in particular, identically zero modulo g. Thus, we have the polynomial identity

$$Q(z, f(z), f^2(z), \ldots, f^{h^{\ell-1}}(z)) \equiv 0 \mod g(z),$$

and by expanding the identity, that

$$\sum_{j=(j_1,\ldots,j_\ell)} (Q_j(z) \bmod g(z)) \cdot (f(z))^{j_1} (f^h(z))^{j_2} \cdots (f^{h^{\ell-1}}(z))^{j_\ell} \equiv 0,$$

which simplifies to the identity

$$\sum_{j=(j_1,\ldots,j_\ell)} (Q_j(z) \bmod g(z)) \cdot (f(z))^{j_1 + j_2 h + \cdots + j_\ell h^{\ell-1}} \equiv 0. \tag{2.7}$$

Consider the degree n field extension $\mathbb{F} = \mathbb{F}_q[z]/g(z)$ of $\mathbb{F}_q$, that is isomorphic to the set of $\mathbb{F}_q$-polynomials of degree smaller than n. Under this notation, for every j let $\alpha_j \in \mathbb{F}$ to be the extension field element corresponding to the $\mathbb{F}_q$-polynomial $(Q_j(z) \bmod g(z))$. Recall that, by our choice of Q, at least one of the α_j's is nonzero, and (2.7) implies that the nonzero univariate $\mathbb{F}$-polynomial

$$\sum_{j=(j_1,\ldots,j_\ell)} \alpha_j z^{j_1 + j_2 h + \cdots + j_\ell h^{\ell-1}}$$

has f, regarded as an element of $\mathbb{F}$, as one of its zeros. The degree of this polynomial is less than K and thus it can have less than K zeros. Thus we conclude that $|S| < K$ and get the desired contradiction. □

By a careful choice of the parameters h and q in the above construction (roughly, $h \approx (2nk/\epsilon)^{1/\alpha}$ and $q \approx h^{1+\alpha}$ for arbitrary constant $\alpha > 0$ and error ϵ), Guruswami et al. derived the following corollary of the above theorem:

Theorem 2.22. [78] *For all constants $\alpha \in (0,1)$ and every $k \leq n \in \mathbb{N}$, $\epsilon > 0$ there is an explicit strong (k, ϵ) lossless condenser with seed length $d = (1 + 1/\alpha)\log(nk/\epsilon) + O(1)$ and output length $m = d + (1+\alpha)k$.* □

Using a straightforward observation, we slightly strengthen this result and show that in fact the parameters can be set up in such a way that the resulting lossless condenser becomes linear. Linearity of the condenser is a property that is particularly useful for the results obtained in Chapter 5.

Corollary 2.23. *Let p be a fixed prime power and $\alpha > 0$ be an arbitrary constant. Then, for parameters $n \in \mathbb{N}$, $k \leq n \log p$, and $\epsilon > 0$, there is an explicit strong $(\leq k, \epsilon)$ lossless condenser $f\colon \mathbb{F}_p^n \times \{0,1\}^d \to \mathbb{F}_p^m$ with seed length $d \leq (1 + 1/\alpha)(\log(nk/\epsilon) + O(1))$ and output length satisfying*[3] *$m \log p \leq d + (1+\alpha)k$. Moreover, f is a linear function (over $\mathbb{F}_p$) for every fixed choice of the seed.*

[3] All unsubscripted logarithms are to the base 2.

Proof. We set up the parameters of the condenser C given by Construction 2.2 and apply Theorem 2.21. The range of the parameters is mostly similar to what chosen in the original result of Guruswami et al. [78].

Letting $h_0 := (2p^2nk/\epsilon)^{1/\alpha}$, we take h to be an integer power of p in range $[h_0, ph_0]$. Also, let $\ell := \lceil k/\log h \rceil$ so that the condition $\ell \geq k/\log h$ required by Theorem 2.21 is satisfied. Finally, let $q_0 := nh\ell/\epsilon$ and choose the field size q to be an integer power of p in range $[q_0, pq_0]$.

We choose the input length of the condenser C to be equal to n. Note that C is defined over $\mathbb{F}_q$, and we need a condenser over $\mathbb{F}_p$. Since q is a fixed parameter, we can ensure that $q \geq p$ (for large enough n), so that $\mathbb{F}_p$ is a subfield of $\mathbb{F}_q$. For $x \in \mathbb{F}_p^n$ and $z \in \{0,1\}^d$, let $y := C(x,y) \in \mathbb{F}_q^\ell$, where x is regarded as a vector over the extension $\mathbb{F}_q$ of $\mathbb{F}_p$. We define the output of the condenser $f(x,z)$ to be the vector y regarded as a vector of length $\ell \log_p q$ over $\mathbb{F}_p$ (by expanding each element of $\mathbb{F}_q$ as a vector of length $\log_p q$ over $\mathbb{F}_p$). It can be clearly seen that f is a strong $(\leq k, \epsilon)$-condenser if C is.

By Theorem 2.21, C is a strong lossless condenser with error upper bounded by

$$\frac{(n-1)(h-1)\ell}{q} \leq \frac{nh\ell}{q_0} = \epsilon.$$

It remains to analyze the seed length and the output length of the condenser. For the output length of the condenser, we have

$$m \log p = \ell \log q \leq (1 + k/\log h) \log q \leq d + k(\log q)/(\log h),$$

where the last inequality is due to the fact that we have $d = \lceil \log q \rceil$. Thus in order to show the desired upper bound on the output length, it suffices to show that $\log q \leq (1+\alpha) \log h_0$. We have

$$\log q \leq \log(pq_0) = \log(pnh\ell/\epsilon) \leq \log h_0 + \log(p^2 n\ell/\epsilon)$$

and our task is reduced to showing that $p^2 n\ell/\epsilon \leq h_0^\alpha = 2p^2nk/\epsilon$. But this bound is obviously valid by the choice of $\ell \leq 1 + k/\log h$.

The seed length is $d = \lceil \log q \rceil$ for which we have

$$\begin{aligned} d &\leq \log q + 1 \leq \log q_0 + O(1) \\ &\leq \log(nh_0\ell/\epsilon) + O(1) \\ &\leq \log(nh_0 k/\epsilon) + O(1) \\ &\leq \log(nk/\epsilon) + \frac{1}{\alpha} \log(2p^2nk/\epsilon) \\ &\leq \big(1 + \frac{1}{\alpha}\big)(\log(nk/\epsilon) + O(1)) \end{aligned}$$

as desired.

Since $\mathbb{F}_q$ has a fixed characteristic, an efficient deterministic algorithm for representation and manipulation of the field elements is available [138] which implies that the condenser is polynomial-time computable and is thus explicit.

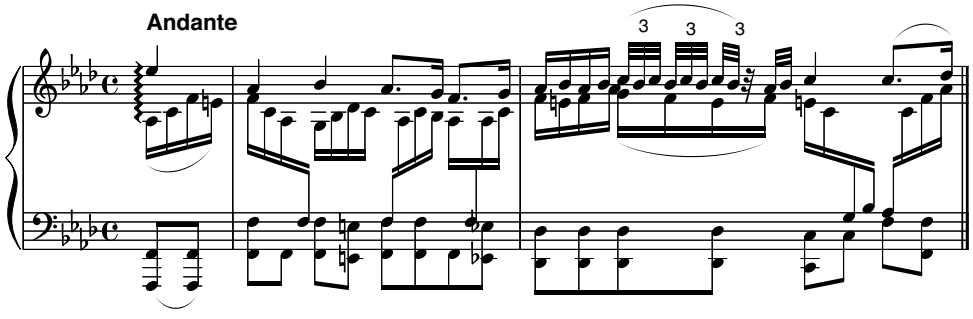
Moreover, since h is taken as an integer power of p and $\mathbb{F}_q$ is an extension of $\mathbb{F}_p$, for any choice of polynomials $F, F', G \in \mathbb{F}_q[X]$, subfield elements $a, b \in \mathbb{F}_p$, and integer $i \geq 0$, we have

$$(aF + bF')^{h^i} \equiv aF^{h^i} + bF'^{h^i} \mod G,$$

meaning that raising a polynomial to power h^i is an $\mathbb{F}_p$-linear operation. Therefore, the mapping C that defines the condenser (Construction 2.2) is $\mathbb{F}_p$-linear for every fixed seed. This in turn implies that the final condenser f is linear, as claimed. □

Guruswami et al. used the lossless condenser above as an intermediate building block for construction of an extractor that is optimal up to constant factors and extracts almost the entire source entropy. Namely, they proved the following result that will be useful for us in later chapters.

Theorem 2.24. [78] *For all positive integers $n \geq k$ and all $\epsilon > 0$, there is an explicit strong (k, ϵ)-extractor* $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \mathbb{F}_2^m$ *with* $m = k - 2\log(1/\epsilon) - O(1)$ *and* $d = \log n + O(\log k \cdot \log(k/\epsilon))$. □



Johann Sebastian Bach (1685–1750): Chorale Prelude in F minor BWV 639 "Ich ruf zu dir, Herr Jesu Christ". Piano transcription by Ferruccio Busoni (1866–1924).

"Music is meaningless noise unless it touches a receiving mind."

— *Paul Hindemith*

Chapter 3

The Wiretap Channel Problem

Suppose that Alice wants to send a message to Bob through a communication channel, and that the message is *partially* observable by an intruder. This scenario arises in various practical situations. For instance, in a packet network, the sequence transmitted by Alice through the channel can be fragmented into small packets at the source and/or along the way and different packets might be routed through different paths in the network in which an intruder may have compromised some of the intermediate routers. An example that is similar in spirit is furnished by transmission of a piece of information from multiple senders to one receiver, across different delivery media, such as satellite, wireless, and/or wired networks. Due to limited resources, a potential intruder may be able to observe only a fraction of the lines of transmission, and hence only partially observe the message. As another example, one can consider secure storage of data on a distributed medium that is physically accessible in parts by an intruder, or a sensitive file on a hard drive that is erased from the file system but is only partially overwritten with new or random information, and hence, is partially exposed to a malicious party.

An obvious approach to solve this problem is to use a secret key to encrypt the information at the source. However, almost all practical cryptographic techniques are shown to be secure only under unproven hardness assumptions and the assumption that the intruder possesses bounded computational power. This might be undesirable in certain situations. Moreover, the key agreement problem has its own challenges.

In the problem that we consider in this chapter, we assume the intruder to be *information theoretically* limited, and our goal will be to employ this limitation and construct a protocol that provides unconditional, information-theoretic security, even in the presence of a computationally unbounded adversary.

The problem described above was first formalized by Wyner [165] and

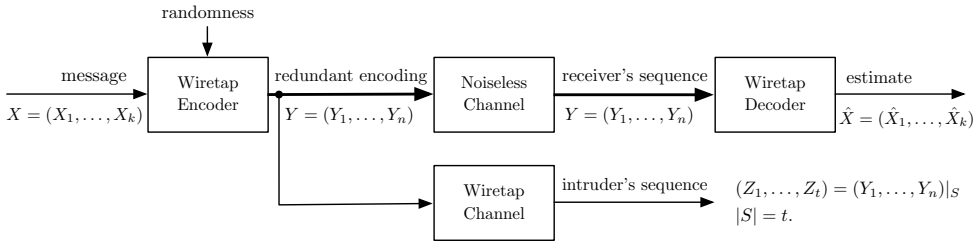


Figure 3.1: The Wiretap II Problem.

subsequently by Ozarow and Wyner [116] as an information-theoretic problem. In its most basic setting, this problem is known as the *wiretap II problem* (the description given here follows from [116]):

> Consider a communication system with a source which outputs a sequence $X = (X_1, \ldots, X_m)$ in $\{0,1\}^m$ uniformly at random. A randomized algorithm, called the encoder, maps the output of the source to a binary string $Y \in \{0,1\}^n$. The output of the encoder is then sent through a noiseless channel (called *the direct channel*) and is eventually delivered to a decoder[1] D which maps Y back to X. Along the way, an intruder arbitrarily picks a subset $S \subseteq [n] := \{1, \ldots, n\}$ of size $t \leq n$, and is allowed to observe[2] $Z := Y|_S$ (through a so-called *wiretap channel*), i.e., Y on the coordinate positions corresponding to the set S. The goal is to make sure that the intruder learns as little as possible about X, regardless of the choice of S.

The system defined above is illustrated in Figure 3.1. The security of the system is defined by the following conditional entropy, known as "equivocation":

$$\Delta := \min_{S:\ |S|=t} H(X|Z).$$

When $\Delta = H(X) = m$, the intruder obtains no information about the transmitted message and we have *perfect privacy* in the system. Moreover, when $\Delta \to m$ as $m \to \infty$, we call the system *asymptotically perfectly* private. These two cases correspond to what is known in the literature as "strong secrecy". A weaker requirement (known as "weak secrecy") would be to have $m - \Delta = o(m)$.

[1] Ozarow and Wyner also consider the case in which the decoder errs with negligible probability, but we are going to consider only error-free decoders.

[2] For a vector $x = (x_1, x_2, \ldots, x_n)$ and a subset $S \subseteq [n]$, we denote by $x|_S$ the vector of length $|S|$ that is obtained from x by removing all the coordinates x_i, $i \notin S$.

Remark 3.1. The assumption that X is sampled from a uniformly random source should not be confused with the fact that Alice is transmitting *one particular* message to Bob that is fixed and known to her before the transmission. In this case, the randomness of X in the model captures the *a priori* uncertainty about X for the *outside world*, and in particular the intruder, but not the transmitter.

As an intuitive example, suppose that a random key is agreed upon between Alice and a trusted third party, and now Alice wishes to securely send her particular key to Bob over a wiretapped channel. Or, assume that Alice wishes to send an audio stream to Bob that is encoded and compressed using a conventional audio encoding method.

Furthermore, the particular choice of the distribution on X as a uniformly random sequence will cause no loss of generality. If the distribution of X is publicly known to be non-uniform, the transmitter can use a suitable source-coding scheme to compress the source to its entropy prior to the transmission, and ensure that from the intruder's point of view, X is uniformly distributed. On the other hand, it is also easy to see that if a protocol achieves perfect privacy under uniform message distribution, it achieves perfect privacy under any other distribution as well.

3.1 The Formal Model

The model that we will be considering in this chapter is motivated by the original wiretap channel problem but is more stringent in terms of its security requirements. In particular, instead of using Shannon entropy as a measure of uncertainty, we will rely on statistical indistinguishability which is a stronger measure that is more widely used in cryptography.

Definition 3.2. Let Σ be a set of size q, m and n be positive integers, and $\epsilon, \gamma > 0$. A $(t, \epsilon, \gamma)_q$-resilient wiretap protocol of block length n and message length m is a pair of functions $E\colon \Sigma^m \times \{0,1\}^r \to \Sigma^n$ (the encoder) and $D\colon \Sigma^n \to \Sigma^m$ (the decoder) that are computable in time polynomial in m, such that

(a) (Decodability) For all $x \in \Sigma^m$ and all $z \in \{0,1\}^r$ we have $D(E(x,z)) = x$,

(b) (Resiliency) Let $X \sim \mathcal{U}_{\Sigma^m}$, $R \sim \mathcal{U}_r$, and $Y = E(X,R)$. For a set $S \subseteq [n]$ and $w \in \Sigma^{|S|}$, let $\mathcal{X}_{S,w}$ denote the distribution of X conditioned on the event $Y|_S = w$. Define the set of *bad observations* as

$$B_S := \{w \in \Sigma^{|S|} \mid \mathsf{dist}(\mathcal{X}_{S,w}, \mathcal{U}_{\Sigma^m}) > \epsilon\},$$

where $\mathsf{dist}(\cdot,\cdot)$ denotes the statistical distance between two distributions. Then we require that for every $S \subseteq [n]$ of size at most t, $\Pr[Y|_S \in B_S] \leq \gamma$, where the probability is over the randomness of X and R.

The *encoding* of a vector $x \in \Sigma^k$ is accomplished by choosing a vector $Z \in \{0,1\}^r$ uniformly at random, and calculating $E(x,Z)$. The quantities $R := m/n$, ϵ, and γ are called the *rate*, the *error*, and the *leakage* of the protocol, respectively. Moreover, we call $\delta := t/n$ the *(relative) resilience* of the protocol.

The decodability condition ensures that the functions E and D are a *matching* encoder/decoder pair, while the resiliency conditions ensures that the intruder learns almost nothing about the message from his observation.

In our definition, the imperfection of the protocol is captured by the two parameters ϵ and γ. When $\epsilon = \gamma = 0$, the above definition coincides with the original wiretap channel problem for the case of perfect privacy.

When $\gamma = 0$, we will have a *worst-case* guarantee, namely, that the intruder's views of the message before and after his observation are statistically close, *regardless* of the outcome of the observation.

The protocol remains interesting even when γ is positive but sufficiently small. When $\gamma > 0$, a particular observation might potentially reveal to the intruder a lot of information about the message. However, a negligible γ will ensure that such a bad event (or *leakage*) happens only with negligible probability.

All the constructions that we will study in this chapter achieve zero leakage (i.e., $\gamma = 0$), except for the general result in Section 3.7.3 for which a nonzero leakage is inevitable.

The significance of zero-leakage protocols is that they assure *adaptive* resiliency in the *weak* sense introduced in [47] for exposure-resilient functions: if the intruder is given the encoded sequence as an oracle that he can adaptively query at up to t coordinates (that is, the choice of each query may depend on the outcome of the previous queries), and is afterwards presented with a challenge which is either the original message or an independent uniformly chosen random string, he will not be able to distinguish between the two cases.

In general, it is straightforward to verify that our model can be used to solve the original wiretap II problem, with $\Delta \geq m(1-\epsilon-\gamma)$:

Lemma 3.3. *Suppose that (E,D) is an encoder/decoder pair as in Definition 3.2. Then using E and D in the wiretap II problem attains an equivocation*

$$\Delta \geq m(1-\epsilon-\gamma).$$

Proof. Let $W := Y|_S$ be the intruder's observation, and denote by W' the set of *good* observations, namely,

$$W' := \{w \in \Sigma^t \colon \mathsf{dist}(\mathcal{X}_{S,w}, \mathcal{U}_{\Sigma^m}) \leq \epsilon\}.$$

Denote by $H(\cdot)$ the Shannon entropy in d-ary symbols. Then we will have

$$\begin{aligned} H(X|W) &= \sum_{w\in\Sigma^t} \Pr(W=w)H(X|W=w) \\ &\geq \sum_{w\in W'} \Pr(W=w)H(X|W=w) \\ &\overset{\text{(a)}}{\geq} \sum_{w\in W'} \Pr(W=w)(1-\epsilon)m \overset{\text{(b)}}{\geq} (1-\gamma)(1-\epsilon)m \geq (1-\gamma-\epsilon)m. \end{aligned}$$

The inequality (a) follows from the definition of W' combined with Proposition 3.30 in the appendix, and (b) by the definition of leakage parameter. □

Hence, we will achieve asymptotically perfect privacy when $\epsilon+\gamma = o(1/m)$. For all the protocols that we present in this chapter this quantity will be superpolynomially small; that is, smaller than $1/m^c$ for every positive constant c (provided that m is large enough).

3.2 Review of the Related Notions in Cryptography

There are several interrelated notions in the literature on Cryptography and Theoretical Computer Science that are also closely related to our definition of the wiretap protocol (Definition 3.2). These are *resilient functions (RF)* and *almost perfect resilient functions (APRF)*, *exposure-resilient functions (ERF)*, and *all-or-nothing transforms (AONT)* (cf. [22, 36, 61, 62, 96, 126, 143] and [45] for a comprehensive account of several important results in this area).

The notion of resilient functions was introduced in [11] (and also [158] as the *bit-extraction problem*). A deterministic polynomial-time computable function $f\colon \{0,1\}^n \to \{0,1\}^m$ is called t-resilient if whenever any t bits of the its input are arbitrarily chosen by an adversary and the rest of the bits are chosen uniformly at random, then the output distribution of the function is (close to) uniform. APRF is a stronger variation where the criterion for uniformity of the output distribution is defined with respect to the ℓ_∞ (i.e., point-wise distance of distributions) rather than ℓ_1. This stronger requirement allows for an "adaptive security" of APRFs.

ERFs, introduced in [22], are similar to resilient functions except that the entire input is chosen uniformly at random, and the view of the adversary from the output remains (close to) uniform even after observing any t input bits of his choice.

ERFs and resilient functions are known to be useful in a scenario similar to the wiretap channel problem where the two parties aim to agree on *any* random string, for example a session key (Alice generates x uniformly at random which she sends to Bob, and then they agree on the string $f(x)$). Here no control

on the content of the message is required, and the only goal is that at the end of the protocol the two parties agree on any random string that is uniform even conditioned on the observations of the intruder. Hence, Definition 3.2 of a wiretap protocol is more stringent than that of resilient functions, since it requires the existence and efficient computability of the encoding function E that provides a control over the content of the message.

Another closely related notion is that of all-or-nothing transforms, which was suggested in [126] for protection of block ciphers. A randomized polynomial-time computable function $f\colon \{0,1\}^m \to \{0,1\}^n$, $(m \leq n)$, is called a (statistical, non-adaptive, and secret-only) t-AONT with error ϵ if it is efficiently invertible and for every $S \subseteq [n]$ such that $|S| \leq t$, and all $x_1, x_2 \in \{0,1\}^m$ we have that the two distributions $f(x_1)|_S$ and $f(x_2)|_S$ are ϵ-close.

An AONT with $\epsilon = 0$ is called perfect. It is easy to see that perfectly private wiretap protocols are equivalent to perfect adaptive AONTs. It was shown in [47] that such functions can not exist (with positive, constant rate) when the adversary is allowed to observe more than half of the encoded bits. A similar result was obtained in [36] for the case of perfect linear RFs.

As pointed out in [47], AONTs can be used in the original scenario of Ozarow and Wyner's wiretap channel problem. However, the best known constructions of AONTs can achieve rate-resilience trade-offs that are far from the information-theoretic optimum (see Figure 3.2).

While an AONT requires indistinguishability of intruder's view for every fixed pair (x_1, x_2) of messages, the relaxed notion of *average-case* AONT requires the *expected* distance of $f(x_1)|_S$ and $f(x_2)|_S$ to be at most ϵ for a uniform random message pair. Hence, for a negligible ϵ, the distance will be negligible for all but a negligible fraction of message pairs. Up to a loss in parameters, wiretap protocols are equivalent to average case AONTs:

Lemma 3.4. *Let (E, D) be an encoding/decoding pair for a $(t, \epsilon, \gamma)_2$-resilient wiretap protocol. Then E is an average-case t-AONT with error at most $2(\epsilon + \gamma)$.*

Conversely, an average-case t-AONT with error η^2 can be used as a (t, η, η)-resilient wiretap encoder.

Proof. Consider a $(t, \epsilon, \gamma)_2$-resilient wiretap protocol as in Definition 3.2, and accordingly, let the random variable $Y = E(X, R)$ denote the encoding of X with a random seed R. For a set $S \subseteq [n]$ of size at most t, denote by $W := Y|_S$ the intruder's observation.

The resiliency condition implies that, the set of bad observations B_S has a probability mass of at most γ and hence, the expected distance $\mathsf{dist}(X|W, X)$ taken over the distribution of W is at most $\epsilon + \gamma$. Now we can apply Proposition 3.31 to the jointly distributed pair of random variables (W, X), and conclude that the expected distance $\mathsf{dist}(W|X, W)$ over the distribution of X (which is uniform) is at most $\epsilon + \gamma$. This implies that the encoder is an average-case t-AONT with error at most $2(\epsilon + \gamma)$.

Conversely, the same argument combined with Markov's bound shows that an average-case t-AONT with error η^2 can be seen as (t, η, η)-resilient wiretap protocol. □

Note that the converse direction does not guarantee zero leakage, and hence, zero leakage wiretap protocols are in general stronger than average-case AONTs. An average-case to worst-case reduction for AONTs was shown in [22] which, combined with the above lemma, can be used to show that any wiretap protocol can be used to construct an AONT (at the cost of a rate loss).

A simple *universal* transformation was proposed in [22] to obtain an AONT from any ERF, by one-time padding the message with a random string obtained from the ERF. In particular, given an ERF $f\colon \{0,1\}^n \to \{0,1\}^m$, the AONT $g\colon \{0,1\}^m \to \{0,1\}^{m+n}$ is defined as $g(x) := (r, x + f(r))$, where $r \in \{0,1\}^n$ is chosen uniformly at random. Hence, the ERF is used to one-time pad the message with a random secret string.

This construction can also yield a wiretap protocol with zero leakage. However, it has the drawback of significantly weakening the rate-resilience trade-off. Namely, even if an information theoretically optimal ERF is used in this reduction, the resulting wiretap protocol will only achieve half the optimal rate (see Figure 3.2). This is because the one-time padding strategy necessarily requires a random seed that is at least as long as the message itself, even if the intruder is restricted to observe only a small fraction of the transmitted sequence. Hence the rate of the resulting AONT cannot exceed $1/2$, and it is not clear how to improve this universal transformation to obtain a worst-case AONT using a shorter seed.

The main focus of this chapter is on asymptotic trade-offs between the rate R and the resilience δ of an asymptotically perfectly private wiretap protocol. For applications in cryptography, e.g., the context of ERFs or AONTs, it is typically assumed that the adversary learns *all* but a small number of the bits in the encoded sequence, and the incurred *blow-up* in the encoding is not as crucially important, as long as it remains within a reasonable range. On the other hand, as in this chapter we are motivated by the wiretap channel problem which is a communication problem, optimizing the transmission *rate* will be the most important concern for us. We will focus on the case where the fraction δ of the symbols observed by the intruder is an arbitrary constant below 1, which is the most interesting range in our context. However, some of our constructions work for sub-constant $1 - \delta$ as well.

Following [116], it is easy to see that, for resilience δ, an information-theoretic bound $R \leq 1 - \delta + o(1)$ must hold. Lower bounds for R in terms of δ have been studied by a number of researchers.

For the case of perfect privacy (where the equivocation Δ is equal to the message length m), Ozarow and Wyner [116] give a construction of a wiretap protocol using linear error-correcting codes, and show that the existence of an

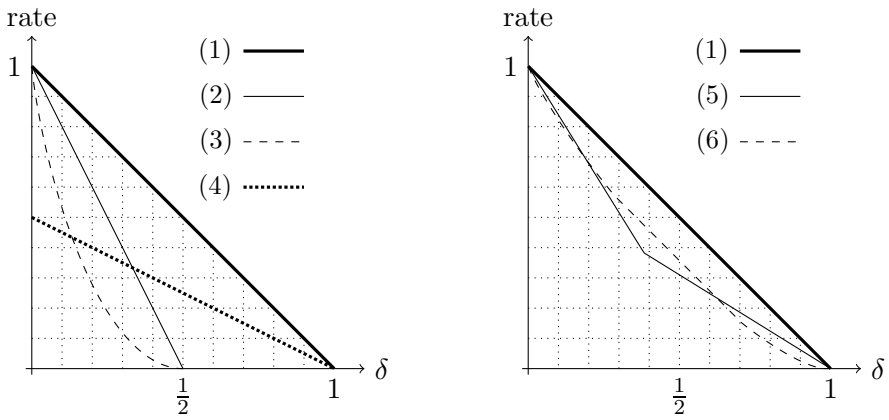


Figure 3.2: A comparison of the rate vs. resilience trade-offs achieved by the wiretap protocols for the binary alphabet (left) and larger alphabets (right, in this example of size 64). (1) Information-theoretic bound, attained by Theorem 3.25; (2) The bound approached by [96]; (3) Protocol based on best non-explicit binary linear codes [68, 157]; (4) AONT construction of [22], assuming that the underlying ERF is optimal; (5) Random walk protocol of Corollary 3.19; (6) Protocol based on the best known explicit [154] and non-explicit [68, 157] linear codes.

$[n, k, d]_q$-code implies the existence of a perfectly private, $(d-1, 0, 0)_q$-resilient wiretap protocol with message length k and block length n (thus, rate k/n).

As a result, the so-called Gilbert-Varshamov bound on the rate-distance trade-offs of linear codes (see Chapter 6) implies that, asymptotically, $R \geq 1 - h_q(\delta)$, where h_q is the q-ary entropy function defined as

$$h_q(x) := x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x).$$

If $q \geq 49$ is a square, the bound can be further improved to $R \geq 1-\delta-1/(\sqrt{q}-1)$ using Goppa's algebraic-geometric codes [72, 154]. In these protocols, the encoder can be seen as an adaptively secure, perfect AONTs and the decoder is an adaptive perfect RF.

Moving away from perfect to asymptotically perfect privacy, it was shown in [96] that for any $\gamma > 0$ there exist binary asymptotically perfectly private wiretap protocols with $R \geq 1 - 2\delta - \gamma$ and exponentially small error[3]. This bound strictly improves the coding-theoretic bound of Ozarow and Wyner for the binary alphabet.

[3]Actually, what is proved in this paper is the existence of t-resilient functions which correspond to decoders in our wiretap setting; however, it can be shown that these functions also possess efficient encoders, so that it is possible to construct wiretap protocols from them.

3.3 Symbol-Fixing and Affine Extractors

Two central notions for our constructions of wiretap protocols in this chapter are symbol-fixing and affine extractors. In this section, we introduce these notions, and study some basic constructions.

Definition 3.5. A d-ary *symbol-fixing source* is an imperfect source of random symbols from an alphabet of size d, that may fix some bounded number of the symbols to unknown values. More precisely, an $(n, k)_d$ *symbol-fixing* source is the distribution of a random variable $X = (X_1, X_2, \ldots, X_n) \in \Sigma^n$, for some set Σ of size d, in which at least k of the coordinates (chosen arbitrarily) are uniformly and independently distributed on Σ and the rest take deterministic values.

When $d = 2$, we will have a *binary* symbol-fixing source, or simply a *bit-fixing* source. In this case $\Sigma = \{0, 1\}$, and the subscript d is dropped from the notation.

The min-entropy of a $(n, k)_d$ symbol-fixing source is $k \log_2 d$ bits. For a d-ary source with $d \neq 2$, it is more convenient to talk about the d-ary entropy of the source, which is k (in d-ary symbols).

Affine sources are natural generalizations of symbol-fixing sources when the alphabet size is a prime power.

Definition 3.6. For a prime power q, an $(n, k)_q$ *affine* source is a distribution on $\mathbb{F}_q^n$ that is uniformly supported on an affine translation of some k-dimensional subspace of $\mathbb{F}_q^n$.

It is easy to see that the q-ary min-entropy of a k-dimensional affine source is k. Due to the restricted structure of symbol-fixing and affine sources, it is possible to construct seedless extractors for such sources:

Definition 3.7. Let Σ be a finite alphabet of size $d > 1$. A function $f : \Sigma^n \to \Sigma^m$ is a (seedless) (k, ϵ)-extractor for symbol-fixing (resp., affine) sources on Σ^n if for every $(n, k)_d$ symbol-fixing (resp., affine) source $\mathcal{X}$, the distribution $E(\mathcal{X})$ is ϵ-close to the uniform distribution $\mathcal{U}_{\Sigma^m}$. The extractor is called explicit if it is deterministic and polynomial-time computable.

We will shortly see simple constructions of zero-error, symbol-fixing and affine extractors using linear functions arising from good error-correcting codes. These extractors achieve the lowest possible error, but however are unable to extract the entire source entropy. Moreover, the affine extractor only works for a "restricted" class of affine sources. For unrestricted affine sources, there are by now various constructions of extractors in the literature. Here we review some notable examples that are most useful for the construction of wiretap protocols that we will discuss in this chapter.

Over large fields, the following affine extractor due to Gabizon and Raz extract almost the entire source entropy:

Theorem 3.8. [65] *There is a constant q_0 such that for any prime power field size q and integers n, k such that $q > \max\{q_0, n^{20}\}$, there is an explicit affine (k, ϵ)-extractor $f\colon \mathbb{F}_q^n \to \mathbb{F}_q^{k-1}$, where $\epsilon < q^{-1/21}$.* □

In this construction, the field size has to be polynomially large in n. When the field size is small (in particular, constant), the task becomes much more challenging. The most challenging case thus corresponds to the binary field $\mathbb{F}_2$, for which an explicit affine extractor was obtained, when the input entropy is a constant fraction of the input length, by Bourgain:

Theorem 3.9. [16] *For every constant $0 < \delta \leq 1$, there is an explicit affine extractor $\mathsf{AExt}\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ for min-entropy δn with output length $m = \Omega(n)$ and error at most $2^{-\Omega(m)}$.* □

Bourgain's construction was recently simplified, improved, and extended to work for arbitrary prime fields by Yehudayoff [167].

An "intermediate" trade-off is recently obtained by DeVos and Gabizon [44], albeit with a short output length. This explicit construction extracts one unbiased bit from any $(n, k)_q$ affine source provided that, for $d := 5n/k$, we have $q > 2d^2$ and the characteristic of the field is larger than d.

3.3.1 Symbol-Fixing Extractors from Linear Codes

The simple theorem below states that linear error-correcting codes can be used to obtain symbol-fixing extractors with zero error.

Theorem 3.10. *Let $\mathcal{C}$ be an $[n, \tilde{k}, d]_q$ code over $\mathbb{F}_q$ and G be a $\tilde{k} \times n$ generator matrix of $\mathcal{C}$. Then, the function $E\colon \mathbb{F}_q^n \to \mathbb{F}_q^{\tilde{k}}$ defined as*[4] *$E(x) := Gx^\top$ is an $(n - d + 1, 0)$-extractor for symbol-fixing sources over $\mathbb{F}_q$.*

Conversely, if a linear function $E\colon \mathbb{F}_q^n \to \mathbb{F}_q^{\tilde{k}}$ is an $(n - d + 1, 0)$-extractor for symbol-fixing sources over $\mathbb{F}_q$, it corresponds to a generator matrix of an $[n, \tilde{k}, d]_q$ code.

Proof. Let $\mathcal{X}$ be a symbol-fixing source with a set $S \subseteq [n]$ of fixed coordinates, where[5] $|S| = d - 1$, and define $\bar{S} := [n] \setminus S$. Observe that, by the Singleton bound, we must have $|\bar{S}| = n - d + 1 \geq \tilde{k}$.

The submatrix of G obtained by removing the columns picked by S must have rank $\tilde{k}$. Since otherwise, the left kernel of this submatrix would be nonzero, meaning that $\mathcal{C}$ has a nonzero codeword that consists of entirely zeros at the $d-1$ positions picked by S, contradicting the assumption that the minimum distance of $\mathcal{C}$ is d. Therefore, the distribution $E(\mathcal{X})$ is supported on a $\tilde{k}$-dimensional affine space on $\mathbb{F}_q^{\tilde{k}}$, meaning that this distribution is uniform.

[4] We typically consider vectors be represented in row form, and use the transpose operator ($x^\top$) to represent column vectors.

[5] If the set of fixed symbols if of size smaller than $d - 1$, the argument still goes through by taking S as an arbitrary set of size $d - 1$ containing all the fixed coordinates.

The converse is straightforward by following the same argument. □

If the field size is large enough; e.g., $q \geq n$, then one can pick $\mathcal{C}$ in the above theorem to be an MDS code (in particular, a Reed-Solomon code) to obtain a $(k, 0)$-extractor for all symbol-fixing sources of entropy k with optimal output length k. However, for a fixed q, negative results on the rate-distance trade-offs of codes (e.g., Hamming, MRRW, and Plotkin bounds) assert that this construction of extractors must inevitably lose some fraction of the entropy of the source. Moreover, the construction would at best be able to extract some constant fraction of the source entropy only if the entropy of the source (in q-ary symbols) is above n/q.

3.3.2 Restricted Affine Extractors from Rank-Metric Codes

In Section 3.7, we will see that affine extractors can be used to construct wiretap schemes for models that are more general than the original Wiretap II problem, e.g., when the direct channel is noisy. For these applications, the extractor needs to additionally have a nice structure that is in particular offered by linear functions.

An obvious observation is that a nontrivial affine extractor cannot be a linear function. Indeed, a linear function $f(x) := \langle \alpha, x \rangle + \beta$, where $\alpha, \beta, x \in \mathbb{F}_q^n$, is constant on the $(n-1)$-dimensional orthogonal subspace of α, and thus, fails to be an extractor for even $(n-1)$-dimensional affine spaces. However, in this section we will see that linear affine extractors can be constructed if the affine source is known to be described by a set of linear constraints whose coefficients lie on a small *sub-field* of the underlying field. Such restricted extractors turn out to be sufficient for some of the applications that we will consider.

Let Q be a prime power. Same as linear codes, an affine subspace on $\mathbb{F}_Q^n$ can be represented by a *generator matrix*, or *parity-check matrix* and a constant shift. That is, a k-dimensional affine subspace $A \subseteq \mathbb{F}_Q^n$ can be described as the image of a linear mapping

$$A := \{xG + \beta \colon x \in \mathbb{F}_Q^k\},$$

where G is a $k \times n$ *generator matrix* of rank k over $\mathbb{F}_Q$, and $\beta \in \mathbb{F}_Q^n$ is a fixed vector. Alternatively, A can be expressed as the translated null-space of a linear mapping

$$A := \{x + \beta \in \mathbb{F}_Q^n \colon Hx^\top = 0\},$$

for an $(n-k) \times n$ *parity check matrix* of rank $n-k$ over $\mathbb{F}_Q$.

Observe that a symbol-fixing source over $\mathbb{F}_q$ with q-ary min-entropy k can be seen as a k-dimensional affine source with a generator matrix of the form $[I \mid \mathbf{0}] \cdot P$, where I is the $k \times k$ identity matrix, $\mathbf{0}$ denotes the $k \times (n-k)$ all-zeros matrix, and P is a permutation matrix. Recall that from Theorem 3.10

we know that for this restricted type of affine sources linear extractors exist. In this section we generalize this idea.

Suppose that $Q = q^m$ for a prime power q so that $\mathbb{F}_Q$ can be regarded as a degree m extension of $\mathbb{F}_q$ (and isomorphic to $\mathbb{F}_{q^m}$). Let A be an affine source over $\mathbb{F}_Q^n$. We will call the affine source $\mathbb{F}_q$-*restricted* if its support can be represented by a generator matrix (or equivalently, a parity check matrix) over $\mathbb{F}_q$.

In this section we introduce an affine extractor that is $\mathbb{F}_Q$-linear and, assuming that m is sufficiently large, extracts from $\mathbb{F}_q$-restricted affine sources. The construction of the extractor is similar to Theorem 3.10, except that instead of an error-correcting code defined over the *Hamming metric*, we will use *rank-metric* codes.

Consider the function $\mathsf{rdist}\colon \mathbb{F}_q^{m\times n} \times \mathbb{F}_q^{m\times n} \to \mathbb{Z}$, where $\mathbb{F}_q^{m\times n}$ denotes the set of $m \times n$ matrices over $\mathbb{F}_q$, defined as $\mathsf{rdist}(A, B) := \mathsf{rank}_q(A - B)$, where rank_q is the matrix rank over $\mathbb{F}_q$. It is straightforward to see that rdist is a metric.

The usual notion of error-correcting codes defined under the Hamming metric can be naturally extended to the rank metric. In particular, a *rank-metric* code $\mathcal{C}$ can be defined as a set of $m \times n$ matrices (known as codewords), whose minimum distance is the minimum rank distance between pairs of codewords.

For $Q := q^m$, there is a natural correspondence between $m\times n$ matrices over $\mathbb{F}_q$ and vectors of length n over $\mathbb{F}_Q$. Consider an isomorphism $\varphi\colon \mathbb{F}_Q \to \mathbb{F}_q^m$ between $\mathbb{F}_Q$ and $\mathbb{F}_q^m$ which maps elements of $\mathbb{F}_Q$ to column vectors of length m over $\mathbb{F}_q$. Then one can define a mapping $\Phi\colon \mathbb{F}_Q^n \to \mathbb{F}_q^{m\times n}$ defined as

$$\Phi(x_1, \ldots, x_n) := [\varphi(x_1) \mid \cdots \mid \varphi(x_n)]$$

to put the elements of $\mathbb{F}_Q^n$ in one-to-one correspondence with $m \times n$ matrices over $\mathbb{F}_q$.

A particular class of rank-metric codes are linear ones. Suppose that $\mathcal{C}$ is a linear $[n, \tilde{k}, \tilde{d}]_Q$ code over $\mathbb{F}_Q$. Then, using $\Phi(\cdot)$, $\mathcal{C}$ can be regarded as a rank-metric code of dimension $\tilde{k}$ over $\mathbb{F}_q^{m\times n}$. In symbols, we will denote such a linear $\tilde{k}$-dimensional rank-metric code as an $[[n, \tilde{k}, d]]_{q^m}$ code, where d is the minimum rank-distance of the code. The rank-distance of a linear rank-metric code turns out to be equal to the minimum rank of its nonzero codewords and obviously, one must have $d \le \tilde{d}$. However, the Hamming distance of $\mathcal{C}$ might turn out to be much larger than its rank distance when regarded as a rank-metric code. In particular, $d \le m$, and thus, d must be strictly smaller than $\tilde{d}$ when the degree m of the field extension is less than $\tilde{d}$.

A counterpart of the Singleton bound in the rank-metric states that, for any $[[n, \tilde{k}, d]]_{q^m}$ code, one must have $d \le n - \tilde{k} + 1$. Rank-metric codes that attain equality exist and are called *maximum rank distance (MRD)* codes. A class of linear rank-metric codes known as *Gabidulin codes* [64] are MRD

and can be thought of as the counterpart of Reed-Solomon codes in the rank metric. In particular, the codewords of a Gabidulin code, seen as vectors over the extension field, are evaluation vectors of bounded-degree *linearized* polynomials rather than arbitrary polynomials as in the case of Reed-Solomon codes. These codes are defined for any choice of $n, \tilde{k}, q, m$ as long as $m \geq n$ and $\tilde{k} \leq n$.

The following is an extension of Theorem 3.10 to restricted affine sources.

Theorem 3.11. *Let $\mathcal{C}$ be an $[[n, \tilde{k}, d]]_{q^m}$ code defined from a code over $\mathbb{F}_Q$ (where $Q := q^m$) with a generator matrix $G \in \mathbb{F}_Q^{\tilde{k} \times n}$. Then the function $E\colon \mathbb{F}_Q^n \to \mathbb{F}_Q^{\tilde{k}}$ defined as $E(x) := Gx^\top$ is an $(n - d + 1, 0)$-extractor for $\mathbb{F}_q$-restricted affine sources over $\mathbb{F}_Q$.*

Conversely, if a linear function $E\colon \mathbb{F}_Q^n \to \mathbb{F}_Q^{\tilde{k}}$ is an $(n-d+1, 0)$-extractor for all $\mathbb{F}_q$-restricted affine sources over $\mathbb{F}_Q$, it corresponds to a generator matrix of an $[[n, \tilde{k}, d]]_{q^m}$ code.

Proof. Consider a restricted affine source $\mathcal{X}$ uniformly supported on an affine subspace of dimension[6] $n - d + 1$

$$X := \{xA + \beta\colon x \in \mathbb{F}_Q^{n-d+1}\},$$

where $A \in \mathbb{F}_q^{(n-d+1)\times n}$ has rank $n - d + 1$, and $\beta \in \mathbb{F}_Q^n$ is a fixed translation. Note that $\tilde{k} \leq n - d + 1$ by the Singleton bound for rank-metric codes.

The output of the extractor is thus uniformly supported on the affine subspace

$$B := \{GA^\top x^\top + G\beta^\top\colon x \in \mathbb{F}_Q^{n-d+1}\} \subseteq \mathbb{F}_Q^{\tilde{k}}.$$

Note that $GA^\top \in \mathbb{F}_Q^{\tilde{k}\times(n-d+1)}$. Our goal is to show that the dimension of B is equal to $\tilde{k}$. Suppose not, then we must have $\mathsf{rank}_Q(GA^\top) < \tilde{k}$. In particular, there is a nonzero $y \in \mathbb{F}_Q^{\tilde{k}}$ such that $yGA^\top = 0$.

Let $Y := \Phi(yG) \in \mathbb{F}_q^{m\times n}$, where $\Phi(\cdot)$ is the isomorphism that maps codewords of $\mathcal{C}$ to their matrix form over $\mathbb{F}_q$. By the distance of $\mathcal{C}$, we know that $\mathsf{rank}_q(Y) \geq d$. Since $m \geq d$, this means that Y has at least d linearly independent rows. On the other hand, we know that the matrix $YA^\top \in \mathbb{F}_q^{\tilde{k}\times(n-d+1)}$ is the zero matrix. Therefore, Y has d independent rows (each in $\mathbb{F}_q^n$) that are all orthogonal to the $n-d+1$ independent rows of A. Since $d+(n-d+1) > n$, this is a contradiction.

Therefore, the dimension of B is exactly $\tilde{k}$, meaning that the output distribution of the extractor is indeed uniform. The converse is straightforward by following a similar line of argument. □

[6] The argument still holds if the dimension of $\mathcal{X}$ is more than $n - d + 1$.

Thus, in particular, we see that generator matrices of MRD codes can be used to construct linear extractors for restricted affine sources that extract the entire source entropy with zero error. This is possible provided that the field size is large enough compared to the field size required to describe the generator matrix of the affine source. Using Gabidulin's rank metric codes, we immediately obtain the following corollary of Theorem 3.11:

Corollary 3.12. *Let q be a prime power. Then for every positive integer n, $k \leq n$, and $Q := q^n$, there is a linear function $f\colon \mathbb{F}_Q^n \to \mathbb{F}_Q^k$ that is a $(k, 0)$-extractor for $\mathbb{F}_q$-restricted affine sources over $\mathbb{F}_Q$.* □

It can be shown using similar proofs that if, in Theorems 3.10 and 3.11, a parity check matrix of the code is used instead of a generator matrix, the resulting linear function would become a lossless $(d-1, 0)$-condenser rather than an extractor. This is in fact part of a more general "duality" phenomenon that is discussed in Section 5.5.

3.4 Inverting Extractors

In this section we will introduce the notion of *invertible extractors* and its connection with wiretap protocols[7]. Later we will use this connection to construct wiretap protocols with good rate-resilience trade-offs.

Definition 3.13. Let Σ be a finite alphabet and f be a mapping from Σ^n to Σ^m. For $\gamma \geq 0$, a function $A\colon \Sigma^m \times \{0,1\}^r \to \Sigma^n$ is called a γ-*inverter* for f if the following conditions hold:

(a) (Inversion) Given $x \in \Sigma^m$ such that $f^{-1}(x)$ is nonempty, for every $z \in \{0,1\}^r$ we have $f(A(x,z)) = x$.

(b) (Uniformity) $A(\mathcal{U}_{\Sigma^m}, \mathcal{U}_r) \sim_\gamma \mathcal{U}_{\Sigma^n}$.

A γ-inverter is called *efficient* if there is a randomized algorithm that runs in worst case polynomial time and, given $x \in \Sigma^m$ and z as a random seed, computes $A(x,z)$. We call a mapping γ-*invertible* if it has an efficient γ-inverter, and drop the prefix γ from the notation when it is zero.

The parameter r in the above definition captures the amount of random bits that the inverter (seen as a randomized algorithm) needs to receive. For

[7] Another notion of invertible extractors was introduced in [46] and used in [48] for a different application (entropic security) that should not be confused with the one we use. Their notion applies to seeded extractors with long seeds that are efficiently invertible bijections for every fixed seed. Such extractors can be seen as a single-step walk on highly expanding graphs that mix in one step. This is in a way similar to the multiple-step random walk used in the seedless extractor of section 3.5, that can be regarded as a single-step walk on the expander graph raised to a certain power.

our applications, no particular care is needed to optimize this parameter and, as long as r is polynomially bounded in n, it is generally ignored.

Remark 3.14. If a function f maps the uniform distribution to a distribution that is ϵ-close to uniform (as is the case for all extractors), then any randomized mapping that maps its input x to a distribution that is γ-close to the uniform distribution on $f^{-1}(x)$ is easily seen to be an $(\epsilon+\gamma)$-inverter for f. In some situations designing such a function might be easier than directly following the above definition.

The idea of random pre-image sampling was proposed in [47] for construction of adaptive AONTs from APRFs. However, they ignored the efficiency of the inversion, as their goal was to show the existence of (not necessarily efficient) information-theoretically optimal adaptive AONTs. Moreover, the strong notion of APRF and a perfectly uniform sampler is necessary for their construction of AONTs. As wiretap protocols are weaker than (worst-case) AONTs, they can be constructed from slightly imperfect inverters as shown by the following lemma.

Lemma 3.15. *Let Σ be an alphabet of size $q > 1$ and $f\colon \Sigma^n \to \Sigma^m$ be a $(\gamma^2/2)$-invertible q-ary (k,ϵ) symbol-fixing extractor. Then, f and its inverter can be seen as a decoder/encoder pair for an $(n-k, \epsilon+\gamma, \gamma)_q$-resilient wiretap protocol with block length n and message length m.*

Proof. Let E and D denote the wiretap encoder and decoder, respectively. Hence, E is the $(\gamma^2/2)$-inverter for f, and D is the extractor f itself. From the definition of the inverter, for every $x \in \Sigma^m$ and every random seed r, we have $D(E(x,r)) = x$. Hence it is sufficient to show that the pair satisfies the resiliency condition.

Let the random variable X be uniformly distributed on Σ^m and the seed $R \in \{0,1\}^r$ be chosen uniformly at random. Denote the encoding of X by $Y := E(X,R)$. Fix any $S \subseteq [n]$ of size at most $n-k$.

For every $w \in \Sigma^{|S|}$, let Y_w denote the set $\{y \in \Sigma^n\colon (y|_S) = w\}$. Note that the sets Y_w partition the space Σ^n into $|\Sigma|^{|S|}$ disjoint sets.

Let $\mathcal{Y}$ and $\mathcal{Y}_S$ denote the distribution of Y and $Y|_S$, respectively. The inverter guarantees that $\mathcal{Y}$ is $(\gamma^2/2)$-close to uniform. Applying Proposition 3.32, we get that

$$\sum_{w\in\Sigma^{|S|}} \Pr[(Y|_S) = w] \cdot \mathsf{dist}((\mathcal{Y}|Y_w), \mathcal{U}_{Y_w}) \le \gamma^2.$$

The left hand side is the expectation of $\mathsf{dist}((\mathcal{Y}|Y_w), \mathcal{U}_{Y_w})$. Denote by W the set of all *bad outcomes* of $Y|_S$, i.e.,

$$W := \{w \in \Sigma^{|S|} \mid \mathsf{dist}((\mathcal{Y}|Y_w), \mathcal{U}_{Y_w}) > \gamma\}.$$

By Markov's inequality, we conclude that

$$\Pr[(Y|_S) \in W] \leq \gamma.$$

For every $w \in W$, the distribution of Y conditioned on the event $(Y|_S) = w$ is γ-close to a symbol-fixing source with $n - |S| \geq k$ random symbols. The fact that D is a symbol-fixing extractor for this entropy and Proposition 3.33 imply that, for any such w, the conditional distribution of $D(Y)|(Y|_S = w)$ is $(\gamma+\epsilon)$-close to uniform. Hence with probability at least $1-\gamma$ the distribution of X conditioned on the outcome of $Y|_S$ is $(\gamma + \epsilon)$-close to uniform. This ensures the resiliency of the protocol. □

By combining Lemma 3.15 and Theorem 3.10 using a Reed-Solomon code, we can obtain a perfectly private, rate-optimal, wiretap protocol for the Wiretap II problem over large alphabets (namely, $q \geq n$), and recover the original result of Ozarow and Wyner[8] [116]:

Corollary 3.16. *For every positive integer n, prime power $q \geq n$, and $\delta \in [0,1)$, there is a $(\delta n, 0, 0)_q$-resilient wiretap protocol with block length n and rate $1-\delta$ that attains perfect privacy.* □

3.5 A Wiretap Protocol Based on Random Walks

In this section we describe a wiretap protocol that achieves a rate R within a constant fraction of the information theoretically optimal value $1-\delta$ (the constant depending on the alphabet size).

To achieve our result, we will modify the symbol-fixing extractor of Kamp and Zuckerman [88], that is based on random walks on expander graphs, to make it efficiently invertible without affecting its extraction properties, and then apply Lemma 3.15 above to obtain the desired wiretap protocol.

Before we proceed, let us briefly review some basic notions and facts related to expander graphs. For a detailed review of the theory of expander graphs, refer to the excellent survey by Hoory, Linial and Wigderson [82], and books [109, 112].

We will be working with directed regular expander graphs that are obtained from undirected graphs by replacing each undirected edge with two directed edges in opposite directions. Let $G = (V, E)$ be a d-regular graph. Then a *labeling* of the edges of G is a function $L\colon V \times [d] \to V$ such that for every $u \in V$ and $t \in [d]$, the edge $(u, L(u,t))$ is in E. The labeling is *consistent* if whenever $L(u,t) = L(v,t)$, then $u = v$. Note that the natural labeling of a Cayley graph (cf. [82]) is in fact consistent.

A *family* of d-regular graphs is an infinite set of d-regular graphs such that for every $N \in \mathbb{N}$, the set contains a graph with at least N vertices. For a

[8] In fact, Ozarow and Wyner use a parity check matrix of an MDS code in their construction, which is indeed a generator matrix for the dual code which is itself MDS.

parameter $c \geq 1$, we will call a family c-dense if there is an $N_0 \in \mathbb{N}$ such that, for every $N \geq N_0$, the family has a graph with at least N and at most cN vertices. We call a family of graphs *constructible* if all the graphs in the family have a consistent labeling that is efficiently computable. That is, there is a uniform, polynomial-time algorithm that, given $N \in \mathbb{N}$ and $i \in [N], j \in [d]$, outputs the label of the jth neighbor of the ith vertex, under a consistent labeling, in the graph in the family that has N vertices (provided that it exists).

Let A denote the normalized adjacency matrix of a d-regular graph G (that is, the adjacency matrix with all the entries divided by d). We denote by λ_G the second largest eigenvalue of A in absolute value. The *spectral gap* of G is given by $1 - \lambda_G$. Starting from a probability distribution p on the set of vertices, represented as a real vector with coordinates index by the vertex set, performing a single-step random walk on G leads to the distribution defined by pA. The following is a well known lemma on the convergence of the distributions resulting from random walks (see [99] for a proof):

Lemma 3.17. *Let $G = (V, E)$ be a d-regular undirected graph, and A be its normalized adjacency matrix. Then for any probability vector p, we have*

$$\|pA - \mathcal{U}_V\|_2 \leq \lambda_G \|p - \mathcal{U}_V\|_2,$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm. □

The extractor of Kamp and Zuckerman [88] starts with a fixed vertex in a large expander graph and interprets the input as the description of a walk on the graph. Then it outputs the label of the vertex reached at the end of the walk. Notice that a direct approach to invert this function will amount to sampling a path of a particular length between a pair of vertices in the graph, uniformly among all the possibilities, which might be a difficult problem for good families of expander graphs[9]. We work around this problem by choosing the starting point of the walk from the input[10]. The price that we pay by doing so is a slightly larger error compared to the original construction of Kamp and Zuckerman that is, asymptotically, of little significance. In particular we show the following:

Theorem 3.18. *Let G be a constructible d-regular graph with d^m vertices and second largest eigenvalue $\lambda_G \geq 1/\sqrt{d}$. Then there exists an explicit invertible*

[9] In fact intractability of the easier problem of finding a loop in certain families of expander graphs forms the underlying basis for a class of cryptographic hash functions (cf. [25]). Even though this easier problem has been solved in [151], uniform sampling of paths seems to be much more difficult.

[10] The idea of choosing the starting point of the walk from the input sequence has been used before in extractor constructions [172], but in the context of seeded extractors for general sources with high entropy.

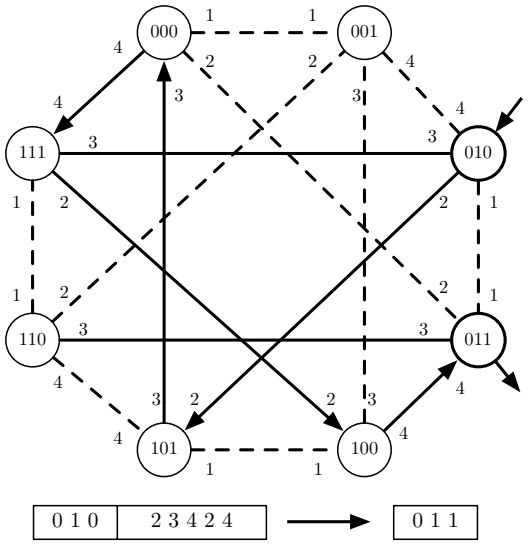


Figure 3.3: The random-walk symbol-fixing extractor.

$(k, 2^{s/2})_d$ *symbol-fixing extractor* $\mathsf{SFExt}\colon [d]^n \to [d]^m$, *such that*

$$s := \begin{cases} m \log d + k \log \lambda_G^2 & \text{if } k \le n - m, \\ (n-k) \log d + (n-m) \log \lambda_G^2 & \text{if } k > n - m. \end{cases}$$

Proof. We first describe the extractor and its inverse. Given an input $(v, w) \in [d]^m \times [d]^{n-m}$, the function SFExt interprets v as a vertex of G and w as the description of a walk starting from v. The output is the index of the vertex reached at the end of the walk. Figure 3.3 depicts the procedure. The 4-regular graph shown in this toy example has 8 vertices labeled with binary sequences of length 3. Edges of the graph are consistently labeled at both endpoints with the set of labels $\{1, 2, 3, 4\}$. The input sequence $(0, 1, 0 \mid 2, 3, 4, 2, 4)$ shown below the graph describes a walk starting from the vertex 010 and following the path shown by the solid arrows. The output of the extractor is the label of the final vertex 011.

The inverter Inv works as follows: Given $x \in [d]^m$, x is interpreted as a vertex of G. Then Inv picks $W \in [d]^{n-m}$ uniformly at random. Let V be the vertex starting from which the walk described by W ends up in x. The inverter outputs (V, W). It is easy to verify that Inv satisfies the properties of a 0-inverter.

Now we show that SFExt is an extractor with the given parameters. We will follow the same line of argument as in the original proof of Kamp and Zuckerman. Let $(x, w) \in [d]^m \times [d]^{n-m}$ be a vector sampled from an $(n, k)_d$ symbol-fixing source, and let $u := \mathsf{SFExt}(x, w)$. Recall that u can be seen as the vertex of G reached at the end of the walk described by w starting from x. Let p_i denote the probability vector corresponding to the walk right after

the ith step, for $i = 0, \ldots, n-m$, and denote by p the uniform probability vector on the vertices of G. Our goal is to bound the error ϵ of the extractor, which is half the ℓ_1 norm of $p_{n-m} - p$.

Suppose that x contains k_1 random symbols and the remaining $k_2 := k - k_1$ random symbols are in w. Then p_0 has the value d^{-k_1} at d^{k_1} of the coordinates and zeros elsewhere, hence

$$\|p_0 - p\|_2^2 = d^{k_1}(d^{-k_1} - d^{-m})^2 + (d^m - d^{k_1})d^{-2m} = d^{-k_1} - d^{-m} \leq d^{-k_1}.$$

Now for each $i \in [n-m]$, if the ith step of the walk corresponds to a random symbol in w the ℓ_2 distance is multiplied by λ_G by Lemma 3.17. Otherwise the distance remains the same due to the fact that the labeling of G is consistent. Hence we obtain $\|p_{n-m} - p\|_2^2 \leq d^{-k_1}\lambda_G^{2k_2}$. Translating this into the ℓ_1 norm by using the Cauchy-Schwarz inequality, we obtain ϵ, namely,

$$\epsilon \leq \frac{1}{2} d^{(m-k_1)/2} \lambda_G^{k_2} < 2^{((m-k_1)\log d + k_2 \log \lambda_G^2)/2}.$$

By our assumption, $\lambda_G \geq 1/\sqrt{d}$. Hence, everything but k_1 and k_2 being fixed, the above bound is maximized when k_1 is minimized. When $k \leq n - m$, this corresponds to the case $k_1 = 0$, and otherwise to the case $k_1 = k - n + m$. This gives us the desired upper bound on ϵ. □

Combining this with Lemma 3.15 and setting up the the right asymptotic parameters, we obtain our protocol for the wiretap channel problem.

Corollary 3.19. *Let $\delta \in [0, 1)$ and $\gamma > 0$ be arbitrary constants, and suppose that there is a constructible family of d-regular expander graphs with spectral gap at least $1 - \lambda$ that is c-dense, for constants $\lambda < 1$ and $c \geq 1$.*

Then, for every large enough n, there is a $(\delta n, 2^{-\Omega(n)}, 0)_d$-resilient wiretap protocol with block length n and rate

$$R = \max\{\alpha(1-\delta), 1 - \delta/\alpha\} - \gamma,$$

where $\alpha := -\log_d \lambda^2$.

Proof. For the case $c = 1$ we use Lemma 3.15 with the extractor SFExt of Theorem 3.18 and its inverse. Every infinite family of graphs must satisfy $\lambda \geq 2\sqrt{d-1}/d$ [114], and in particular we have $\lambda \geq 1/\sqrt{d}$, as required by Theorem 3.18. We choose the parameters $k := (1-\delta)n$ and $m := n(\max\{\alpha(1-\delta), 1 - \delta/\alpha\} - \gamma)$, which gives $s = -\Omega(n)$, and hence, exponentially small error. The case $c > 1$ is similar, but involves technicalities for dealing with lack of graphs of arbitrary size in the family. We will elaborate on this in Appendix 3.A. □

Using explicit constructions of Ramanujan graphs that achieve

$$\lambda \leq 2\sqrt{d-1}/d$$

when $d-1$ is a prime power [100, 110, 119], one can obtain $\alpha \geq 1 - 2/\log d$, which can be made arbitrarily close to one (hence, making the protocol arbitrarily close to the optimal bound) by choosing a suitable alphabet size that does not depend on n. Namely, we have the following result:

Corollary 3.20. *Let $\delta \in [0,1)$ and $\gamma > 0$ be arbitrary constants. Then, there is a positive integer d only depending on γ such that the following holds: For every large enough n, there is a $(\delta n, 2^{-\Omega(n)}, 0)_d$-resilient wiretap protocol with block length n and rate at least $1 - \delta - \gamma$.* □

3.6 Invertible Affine Extractors and Asymptotically Optimal Wiretap Protocols

In this section we will construct a black box transformation for making certain seedless extractors invertible. The method is described in detail for affine extractors, and leads to wiretap protocols with asymptotically optimal rate-resilience trade-offs. Being based on affine extractors, these protocols are only defined for prime power alphabet sizes. On the other hand, the random-walk based protocol discussed in Section 3.5 can be potentially instantiated for an arbitrary alphabet size, though achieving asymptotically sub-optimal parameters (and a positive rate only for an alphabet of size 3 or more).

Modulo some minor differences, the construction can be simply described as follows: A seedless affine extractor is first used to extract a small number of uniform random bits from the source, and the resulting sequence is then used as the seed for a seeded extractor that extracts almost the entire entropy of the source.

Of course, seeded extractors in general are not guaranteed to work if (as in the above construction) their seed is not independent from the source. However, as observed by Gabizon and Raz [65], a *linear* seeded extractor can extract from an affine source if the seed is the outcome of an affine extractor on the source. This idea was formalized in a more general setting by Shaltiel [132].

Shaltiel's result gives a general framework for transforming any seedless extractor (for a family of sources satisfying a certain *closedness* condition) with short output length to one with an almost optimal output length. The construction uses the imperfect seedless extractor to extract a small number of uniform random bits from the source, and will then use the resulting sequence as the seed for a seeded extractor to extract more random bits from the source. For a suitable choice of the seeded extractor, one can use this construction to extract almost all min-entropy of the source.

The closedness condition needed for this result to work for a family $\mathcal{C}$ of sources is that, letting $E(x,s)$ denote the seeded extractor with seed s, for every $\mathcal{X} \in \mathcal{C}$ and every fixed s and y, the distribution $(\mathcal{X}|E(\mathcal{X},s) = y)$ belongs to $\mathcal{C}$. If E is a linear function for every fixed s, the result will be available for

affine sources (since we are imposing a linear constraint on an affine source, it remains an affine source). A more precise statement of Shaltiel's main result is the following:

Theorem 3.21. [132] *Let $\mathcal{C}$ be a class of distributions on $\mathbb{F}_2^n$ and $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^t$ be an extractor for $\mathcal{C}$ with error ϵ. Let $E\colon \mathbb{F}_2^n \times \mathbb{F}_2^t \to \mathbb{F}_2^m$ be a function for which $\mathcal{C}$ satisfies the closedness condition above. Then for every $\mathcal{X} \in \mathcal{C}$, $E(\mathcal{X}, F(\mathcal{X})) \sim_{\epsilon \cdot 2^{t+3}} E(\mathcal{X}, \mathcal{U}_t)$.* □

Recall that a seeded extractor is called *linear* if it is a linear function for every fixed choice of the seed, and that this condition is satisfied by Trevisan's extractor [152]. For our construction, we will use the following theorem implied by the improvement of this extractor due to Raz, Reingold and Vadhan (Theorem 2.20):

Theorem 3.22. [123] *There is an explicit strong linear seeded (k, ϵ)-extractor $\mathsf{Ext}\colon \mathbb{F}_2^n \times \mathbb{F}_2^d \to \mathbb{F}_2^m$ with $d = O(\log^3(n/\epsilon))$ and $m = k - O(d)$.* □

Remark 3.23. We note that our arguments would identically work for any other linear seeded extractor as well, for instance those constructed in [134, 150]. However, the most crucial parameter in our application is the output length of the extractor, being closely related to the rate of the wiretap protocols we obtain. Among the constructions we are aware of, the result quoted in Theorem 3.22 is the best in this regard. Moreover, an affine seeded extractor with better parameters is constructed by Gabizon and Raz [65], but it requires a large alphabet size to work.

Now, having the right tools in hand, we are ready to formally describe our construction of invertible affine extractors with nearly optimal output length. Broadly speaking, the construction follows the abovementioned idea of Shaltiel, Gabizon, and Raz [65, 132] on enlarging the output length of affine extractors, with an additional "twist" for making the extractor invertible. For concreteness, the description is given over the binary field $\mathbb{F}_2$:

Theorem 3.24. *For every constant $\delta \in (0, 1]$ and every $\alpha \in (0, 1)$, there is an explicit invertible affine extractor $D\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ for min-entropy δn with output length $m = \delta n - O(n^\alpha)$ and error at most $O(2^{-n^{\alpha/3}})$.*

Proof. Let $\epsilon := 2^{-n^{\alpha/3}}$, and $t := O(\log^3(n/\epsilon)) = O(n^\alpha)$ be the seed length required by the extractor Ext in Theorem 3.22 for input length n and error ϵ, and further, let $n' := n - t$. Set up Ext for input length n', min-entropy $\delta n - t$, seed length t and error ϵ. Also set up Bourgain's extractor AExt for input length n' and entropy rate δ', for an arbitrary constant $\delta' < \delta$. Then the function F will view the n-bit input sequence as a tuple (s, x), $s \in \mathbb{F}_2^t$ and $x \in \mathbb{F}_2^{n'}$, and outputs $\mathsf{Ext}(x, s + \mathsf{AExt}(x)|_{[t]})$. This is depicted in Figure 3.4.

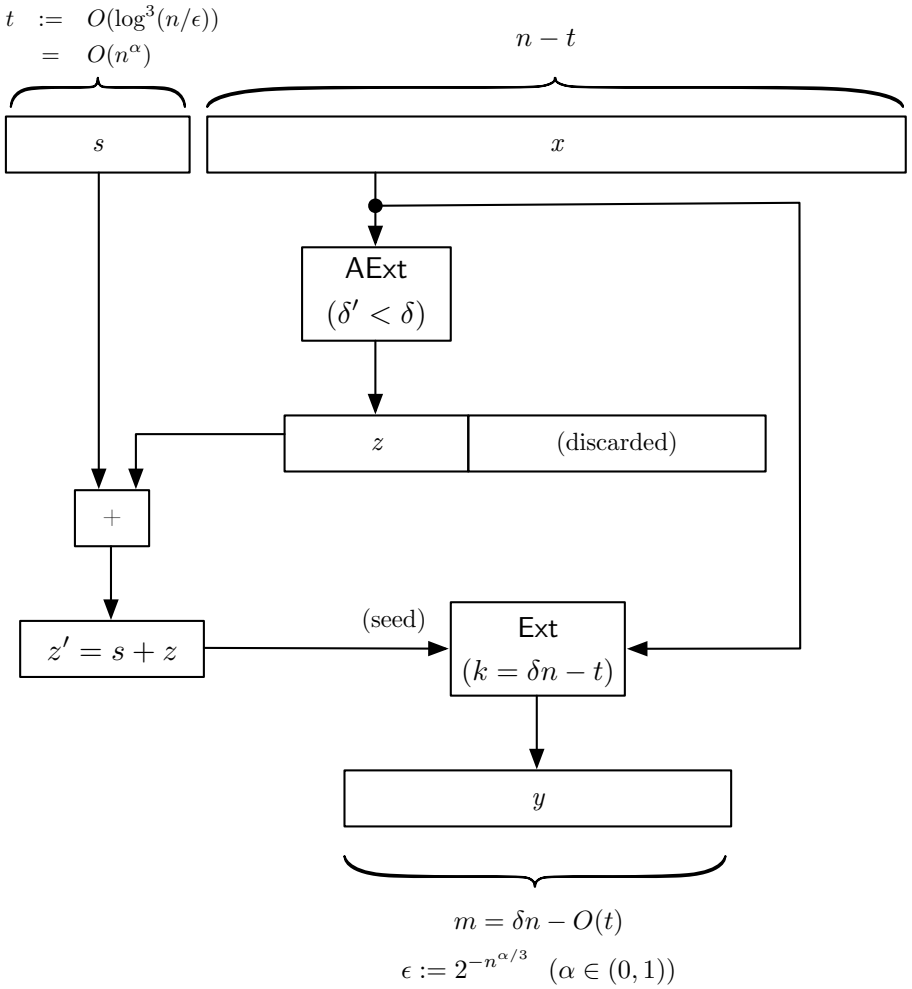


Figure 3.4: Construction of the invertible affine extractor.

First we show that this is an affine extractor. Suppose that $(S, X) \in \mathbb{F}_2^t \times \mathbb{F}_2^{n'}$ is a random variable sampled from an affine distribution with min-entropy δn. The variable S can have an affine dependency on X. Hence, for every fixed $s \in \mathbb{F}_2^t$, the distribution of X conditioned on the event $S = s$ is affine with min-entropy at least $\delta n - t$, which is at least $\delta' n'$ for large enough n. Hence $\mathsf{AExt}(X)$ will be $2^{-\Omega(n)}$-close to uniform by Theorem 3.9. This implies that $\mathsf{AExt}(X)|_{[t]} + S$ can extract t random bits from the affine source with error $2^{-\Omega(n)}$. Combining this with Theorem 3.21, noticing the fact that the class of affine extractors is closed with respect to linear seeded extractors, we conclude that D is an affine extractor with error at most $\epsilon + 2^{-\Omega(n)} \cdot 2^{t+3} = O(2^{-n^{\alpha/3}})$.

Now the inverter works as follows: Given $y \in \mathbb{F}_2^m$, first it picks $Z \in \mathbb{F}_2^t$ uniformly at random. The seeded extractor Ext, given the seed Z is a

linear function $\mathsf{Ext}_Z \colon \mathbb{F}_2^{n'} \to \mathbb{F}_2^m$. Without loss of generality, assume that this function is surjective[11]. Then the inverter picks $X \in \mathbb{F}_2^{n'}$ uniformly at random from the affine subspace defined by the linear constraint $\mathsf{Ext}_Z(X) = y$, and outputs $(Z + \mathsf{AExt}(X)|_{[t]}, X)$. It is easy to verify that the output is indeed a valid preimage of y. To see the uniformity of the inverter, note that if y is chosen uniformly at random, the distribution of (Z, X) will be uniform on $\mathbb{F}_2^n$. Hence $(Z + \mathsf{AExt}(X)|_{[t]}, X)$, which is the output of the inverter, will be uniform. □

In the above construction we are using an affine and a linear seeded extractor as *black boxes*, and hence, they can be replaced by any other extractors as well (the construction will achieve an optimal rate provided that the seeded extractor extracts almost the entire source entropy). In particular, over large fields one can use the affine and seeded extractors given by Gabizon and Raz [65] that work for sub-constant entropy rates as well.

Moreover, for concreteness we described and instantiated our construction over the binary field. Observe that Shaltiel's result, for the special case of affine sources, holds regardless of the alphabet size. Moreover, Trevisan's linear seeded extractor can be naturally extended to handle arbitrary alphabets. Hence, in order to extend our result to non-binary alphabets, it suffices to ensure that a suitable seedless affine extractor that supports the desired alphabet size is available. Bourgain's original result [16] is stated and proved for the binary alphabet; however, it seems that this result can be adapted to work for larger fields as well [17]. Such an extension (along with some improvements and simplifications) is made explicit by Yehudayoff [167].

An affine extractor is in particular, a symbol-fixing extractor. Hence Theorem 3.24, combined with Lemma 3.15 gives us a wiretap protocol with almost optimal parameters:

Theorem 3.25. *Let $\delta \in [0, 1)$ and $\alpha \in (0, 1/3)$ be constants. Then for a prime power $q > 1$ and every large enough n there is a $(\delta n, O(2^{-n^\alpha}), 0)_q$-resilient wiretap protocol with block length n and rate $1 - \delta - o(1)$.* □

3.7 Further Applications

In this section we will sketch some important applications of our technique to more general wiretap problems.

[11] Because the seeded extractor is strong and linear, for most choices of the seed it is a good extractor (by Proposition 2.11), and hence necessarily surjective (if not, one of the output symbols would linearly depend on the others and obviously the output distribution would not be close to uniform). Hence if Ext is not surjective for some seed z, one can replace it by a trivial surjective linear mapping without affecting its extraction properties.

3.7.1 Noisy Channels and Active Intruders

Suppose that Alice wants to transmit a particular sequence to Bob through a noisy channel. She can use various techniques from coding theory to encode her information and protect it against noise. Now what if there is an intruder who can partially observe the transmitted sequence and even manipulate it? Modification of the sequence by the intruder can be regarded in the same way as the channel noise; thus one gets security against active intrusion as a "bonus" by constructing a code that is resilient against noise and passive eavesdropping. There are two natural and modular approaches to construct such a code.

A possible attempt would be to first encode the message using a good error-correcting code and then applying a wiretap encoder to protect the encoded sequence against the wiretapper. However, this will not necessarily keep the information protected against the channel noise, as the combination of the wiretap encoder and decoder does not have to be resistant to noise.

Another attempt is to first use a wiretap encoder and then apply an error-correcting code on the resulting sequence. Here it is not necessarily the case that the information will be kept secure against intrusion anymore, as the wiretapper now gets to observe the bits from the channel-encoded sequence that may reveal information about the original sequence. However, the wiretap protocol given in Theorem 3.25 is constructed from an invertible affine extractor, and guarantees resiliency even if the intruder is allowed to observe arbitrary linear combinations of the transmitted sequence (in this case, the distribution of the encoded sequence subject to the intruder's observation becomes an affine source and thus, the arguments of the proof of Lemma 3.15 remain valid). In particular, Theorem 3.25 holds even if the intruder's observation is allowed to be obtained after applying any arbitrary linear mapping on the output of the wiretap encoder. Hence, we can use the wiretap scheme as an outer code and still ensure privacy against an active intruder and reliability in presence of a noisy channel, provided that the error-correcting code being used as the inner code is linear. This immediately gives us the following result:

Theorem 3.26. *Suppose that there is a q-ary linear error-correcting code with rate r that is able to correct up to a τ fraction of errors (via unique or list decoding). Then for every constant $\delta \in [0,1)$ and $\alpha \in (0,1/3)$ and large enough n, there is a $(\delta n, O(2^{-n^{\alpha}}), 0)_q$-resilient wiretap protocol with block length n and rate $r - \delta - o(1)$ that can also correct up to a τ fraction of errors.* □

The setting discussed above is shown in Figure 3.5. The same idea can be used to protect fountain codes, e.g., LT- [101] and Raptor Codes [137], against wiretappers without affecting the error correction capabilities of the code.

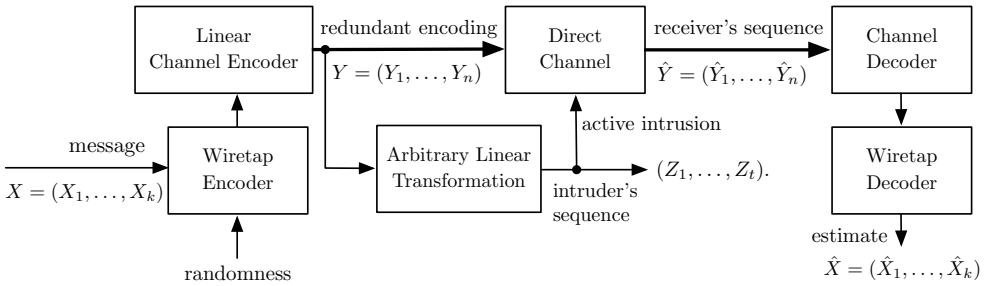


Figure 3.5: Wiretap scheme composed with channel coding. If the wiretap scheme is constructed by an invertible affine extractor, it can guarantee secrecy even in presence of arbitrary linear manipulation of the information. Active intrusion can be defied using an error-correcting inner code.

Obviously this simple composition idea can be used for any type of channel so long as the inner code is linear, at the cost of reducing the total rate by almost δ. Hence, if the inner code achieves the Shannon capacity of the direct channel (in the absence of the wiretapper), the composed code will achieve the capacity of the wiretapped channel, which is less than the original capacity by δ [41].

3.7.2 Network Coding

Our wiretap protocol from invertible affine extractors is also applicable in the more general setting of transmission over networks. A communication network can be modeled as a directed graph, in which nodes represent the network devices and information is transmitted along the edges. One particular node is identified as the *source* and m nodes are identified as *receivers*. The main problem in network coding is to have the source reliably transmit information to the receivers at the highest possible rate, while allowing the intermediate nodes arbitrarily process the information along the way.

Suppose that, in the graph that defines the topology of the network, the min-cut between the source to each receiver is n. It was shown in [4] that the source can transmit information up to rate n (symbols per transmission) to all receivers (which is optimal), and in [94, 97] that linear network coding is in fact sufficient to achieve this rate. That is, the transmission at rate n is possible when the intermediate nodes are allowed to forward packets that are (as symbols over a finite field) linear combinations of the packets that they receive (See [168] for a comprehensive account of these and other relevant results).

A basic example is shown by the *butterfly* network in Figure 3.6. This network consists of a source on the top and two receivers on the bottom, where the min-cut to each receiver is 2. Without processing the incoming

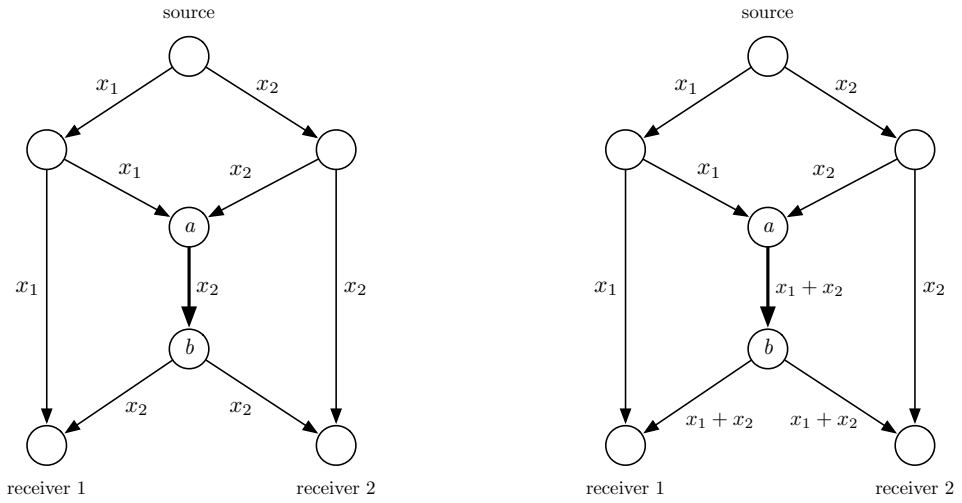


Figure 3.6: Network coding (right), versus unprocessed forwarding (left).

data, as in the left figure, one of the two receivers may receive information at the optimal rate of 2 symbols per transmission (namely, receiver 1 in the figure). However, due to the bottleneck existing in the middle (shown by the thick edge $a \to b$), the other receiver will be forced to receive at an inferior rate of 1 symbol per transmission. However, if linear processing of the information is allowed, node a may combine its incoming information by treating packets as symbols over a finite field and adding them up, as in the right figure. Both receivers may then solve a full-rank system of linear equations to retrieve the original source symbols x_1 and x_2, and thereby achieve the optimal min-cut rate.

Designing wiretap protocols for networks is an important question in network coding, which was first posed by Cai and Yeung [21]. In this problem, an intruder can choose a bounded number, say t, of the edges and eavesdrop all the packets going through those edges. They designed a network code that could provide the optimal multicast rate of $n - t$ with perfect privacy. However this code requires an alphabet size of order $\binom{|E|}{t}$, where E is the set of edges. Their result was later improved in [59] who showed that a random linear coding scheme can provide privacy with a much smaller alphabet size if one is willing to achieve a slightly sub-optimal rate. Namely, they obtain rate $n - t(1 + \epsilon)$ with an alphabet of size roughly $\Theta(|E|^{1/\epsilon})$, and show that achieving the exact optimal rate is not possible with small alphabet size.

El Rouayheb and Soljanin [55] suggested to use the original code of Ozarow and Wyner [116] as an outer code at the source and showed that a careful choice of the network code can provide optimal rate with perfect privacy. However, their code eventually needs an alphabet of size at least $\binom{|E|-1}{t-1} + m$.

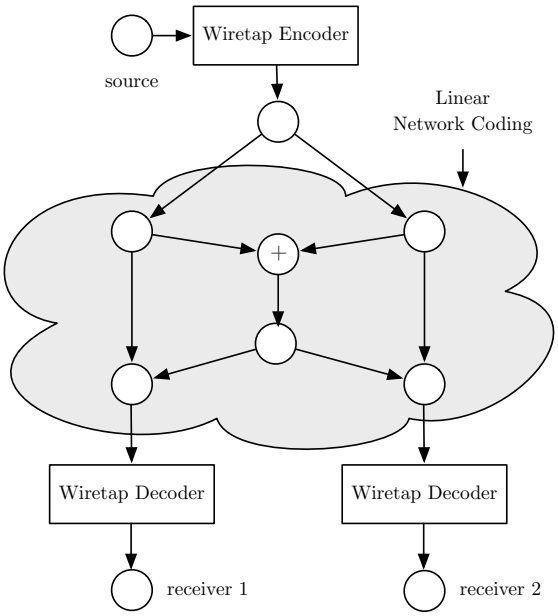


Figure 3.7: Linear network coding with an outer layer of wiretap encoding added for providing secrecy.

Building upon this work, Silva and Kschischang [95] constructed an outer code that provides similar results while leaving the underlying network code unchanged. However, their result comes at the cost of increasing the packet size by a multiplicative factor of at least the min-cut bound, n (or in mathematical terms, the original alphabet size q of the network is enlarged to at least q^n). For practical purposes, this is an acceptable solution provided that an estimate on the min-cut size of the network is available at the wiretap encoder.

By the discussion presented in Section 3.7.1, the rate-optimal wiretap protocol given in Theorem 3.25 stays resilient even in presence of any linear post-processing of the encoded information. Thus, using the wiretap encoder given by this result as an outer-code in the source node, one can construct an asymptotically optimal wiretap protocol for networks that is completely unaware of the network and eliminates all the restrictions in the above results. This is schematically shown in Figure 3.7. Hence, extending our notion of $(t, \epsilon, \gamma)_q$-resilient wiretap protocols naturally to communication networks, we obtain the following:

Theorem 3.27. *Let $\delta \in [0, 1)$ and $\alpha \in (0, 1/3)$ be constants, and consider a network that uses a linear coding scheme over a finite field $\mathbb{F}_q$ for reliably transmitting information at rate R. Suppose that, at each transmission, an intruder can arbitrarily observe up to δR intermediate links in the network.*

Then the source and the receiver nodes can use an outer code of rate $1-\delta-o(1)$ *(obtaining a total rate of* $R(1-\delta)-o(1)$*) which is completely independent of the network, leaves the network code unchanged, and provides almost perfect privacy with error* $O(2^{-R^{\alpha}})$ *and zero leakage over a* q*-ary alphabet.* □

In addition to the above result that uses the invertible affine extractor of Theorem 3.24, it is possible to use other rate-optimal invertiable affine extractors. In particular, observe that the restricted affine extractor of Theorem 3.11 (and in particular, Corollary 3.12) is a linear function (over the extension field) and is thus, obviously has an efficient 0-inverter (since inverting the extractor amounts to solving a system of linear equations). By using this extractor (instantiated with Gabidulin's MRD codes as in Corollary 3.12), we may recover the result of Silva and Kschischang [95] in our framework. More precisely, we have the following result:

Corollary 3.28. *Let* q *be any prime power, and consider a network with minimum cut of size* n *that uses a linear coding scheme over* $\mathbb{F}_q$ *for reliably transmitting information at rate* R*. Suppose that, at each transmission, an intruder can arbitrarily observe up to* δR *intermediate links in the network, for some* $\delta \in [0,1)$*. Then the source and the receiver nodes can use an outer code of rate* $1-\delta$ *over* $\mathbb{F}_{q^n}$ *(obtaining a total rate of* $R(1-\delta)$*) that provides perfect privacy over a* q^n*-ary alphabet.* □

3.7.3 Arbitrary Processing

In this section we consider the erasure wiretap problem in its most general setting, which is still of practical importance. Suppose that the information emitted by the source goes through an arbitrary communication medium and is arbitrarily processed on the way to provide protection against noise, to obtain better throughput, or for other reasons. Now consider an intruder who is able to eavesdrop a bounded amount of information at various points of the channel. One can model this scenario in the same way as the original point-to-point wiretap channel problem, with the difference that instead of observing t arbitrarily chosen bits, the intruder now gets to choose an arbitrary Boolean circuit $\mathcal{C}$ with t output bits (which captures the accumulation of all the intermediate processing) and observes the output of the circuit when applied to the transmitted sequence[12].

Obviously there is no way to guarantee resiliency in this setting, since the intruder can simply choose $\mathcal{C}$ to compute t output bits of the wiretap decoder. However, suppose that in addition there is an auxiliary communication channel between the source and the receiver (that we call the *side channel*) that is separated from the main channel, and hence, the information passed through the two channel do not *blend* together by the intermediate processing.

[12] In fact this models a "harder" problem, as in our problem the circuit $\mathcal{C}$ is given by the communication scheme and not the intruder. Nevertheless, we consider the harder problem.

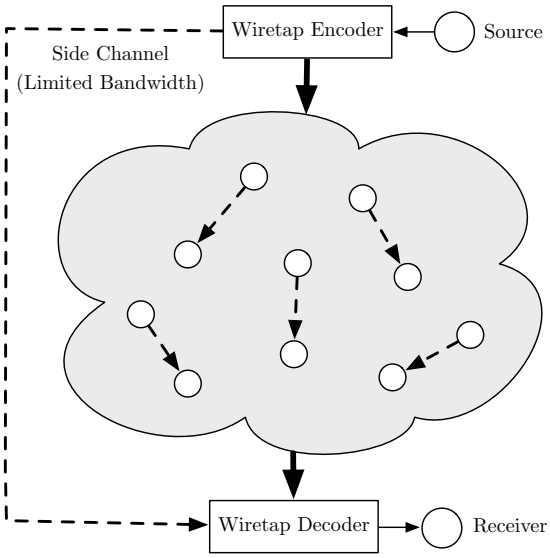


Figure 3.8: The wiretap channel problem in presence of arbitrary intermediate processing. In this example, data is transmitted over a packet network (shown as a cloud) in which some intermediate links (showed by the dashed arrows) are accessible to an intruder.

We call this scenario the *general wiretap problem*, and extend our notion of (t, ϵ, γ)-resilient protocol to this problem, with the slight modification that now the output of the encoder (and the input of the decoder) is a pair of strings $(y_1, y_2) \in \mathbb{F}_2^n \times \mathbb{F}_2^d$, where y_1 (resp., y_2) is sent through the main (resp., side) channel. Now we call $n + d$ the block length and let the intruder choose an arbitrary pair of circuits $(\mathcal{C}_1, \mathcal{C}_2)$, one for each channel, that output a total of t bits, and observe $(\mathcal{C}_1(y_1), \mathcal{C}_2(y_2))$.

The information-theoretic upper bounds for the achievable rates in the original wiretap problem obviously extend to the general wiretap problem as well. Below we show that for the general problem, secure transmission is possible at asymptotically optimal rates even if the intruder intercepts the *entire communication* passing through the side channel (as shown in Figure 3.8).

Similar as before, our idea is to use invertible extractors to construct general wiretap protocols, but this time we use invertible strong seeded extractors. Strong seeded extractors were used in [22] to construct ERFs, and this is exactly what we use as the decoder in our protocol. As the encoder we will use the corresponding inverter, which outputs a pair of strings, one for the extractor's input which is sent through the main channel and another as the seed which is sent through the side channel. Hence we will obtain the following result:

Theorem 3.29. *Let $\delta \in [0, 1)$ be a constant. Then for every $\alpha, \epsilon > 0$, there*

is a $(\delta n, \epsilon, 2^{-\alpha n} + \epsilon)$-resilient wiretap protocol for the general wiretap channel problem that sends n bits through the main channel and $d := O(\log^3(n/\epsilon^2))$ bits through the side channel and achieves rate $1 - \delta - \alpha - O(d/(n + d))$. The protocol is secure even when the entire communication through the side channel is observable by the intruder.

Proof. We will need the following claim in our proof, which is easy to verify using an averaging argument:

Claim. *Let $f\colon \{0,1\}^n \to \{0,1\}^{\delta n}$ be a Boolean function. Then for every $\alpha > 0$, and $X \sim \mathcal{U}_n$, the probability that $f(X)$ has fewer than $2^{n(1-\delta-\alpha)}$ preimages is at most $2^{-\alpha n}$.*

Now, let Ext be the linear seeded extractor of Theorem 3.22, set up for input length n, seed length $d = O(\log^3(n/\epsilon^2))$, min-entropy $n(1-\delta-\alpha)$, and output length $m = n(1-\delta-\alpha)-O(d)$, and error ϵ^2. Then the encoder chooses a seed Z for the extractor uniformly at random and sends it through the side channel.

For the chosen value of Z, the extractor is a linear function, and as before, given a message $x \in \{0,1\}^m$, the encoder picks a random vector in the affine subspace that is mapped by this linear function to x and sends it through the public channel.

The decoder, in turn, applies the extractor to the seed received from the secure channel and the transmitted string. The resiliency of the protocol can be shown in a similar manner as in Lemma 3.15. Specifically, note that by the above claim, with probability at least $1 - 2^{-\alpha n}$, the string transmitted through the main channel, conditioned on the observation of the intruder from the main channel, has a distribution $\mathcal{Y}$ with min-entropy at least $n(1-\delta-\alpha)$. Now in addition suppose that the seed z is entirely revealed to the intruder. As the extractor is strong, with probability at least $1-\epsilon$, z is a *good* seed for $\mathcal{Y}$, meaning that the output of the extractor applied to $\mathcal{Y}$ and seed z is ϵ-close to uniform (by Proposition 2.11), and hence the view of the intruder on the original message remains ϵ-close to uniform. □

We observe that it is not possible to guarantee zero leakage for the general wiretap problem above. Specifically, suppose that $(\mathcal{C}_1, \mathcal{C}_2)$ are chosen in a way that they have a single preimage for a particular output (w_1, w_2). With nonzero probability the observation of the intruder may turn out to be (w_1, w_2), in which case the entire message is revealed. Nevertheless, it is possible to guarantee negligible leakage as the above theorem does. Moreover, when the general protocol above is used for the original wiretap II problem (where there is no intermediate processing involved), there is no need for a separate side channel and the entire encoding can be transmitted through a single channel. Contrary to Theorem 3.25 however, the general protocol will not guarantee zero leakage even for this special case.

3.A Some Technical Details

This appendix is devoted to some technical details that are omitted in the main text of the chapter.

The following proposition quantifies the Shannon entropy of a distribution that is close to uniform:

Proposition 3.30. *Let $\mathcal{X}$ be a probability distribution on a finite set S, $|S| > 4$, that is ϵ-close to the uniform distribution on S, for some $\epsilon \leq 1/4$. Then $H(\mathcal{X}) \geq \log_2 |S|(1-\epsilon)$*

Proof. Let $n := |S|$, and let $f(x) := -x \log_2 x$. The function $f(x)$ is concave, passes through the origin and is strictly increasing in the range $[0, 1/\mathrm{e}]$. From the definition, we have $H(\mathcal{X}) = \sum_{s\in S} f(\Pr_{\mathcal{X}}(s))$. For each term s in this summation, the probability that $\mathcal{X}$ assigns to s is either at least $1/n$, which makes the corresponding term at least $\log_2 n/n$ (due to the particular range of $|S|$ and ϵ), or is equal to $1/n - \epsilon_s$, for some $\epsilon_s > 0$, in which case the term corresponding to s is less than $\log_2 n/n$ by at most $\epsilon_s \log_2 n$ (this follows by observing that the slope of the line connecting the origin to the point $(1/n, f(1/n))$ is $\log_2 n$). The bound on the statistical distance implies that the differences ϵ_s add up to at most ϵ. Hence, the Shannon entropy of $\mathcal{X}$ can be less than $\log_2 n$ by at most $\epsilon \log_2 n$. □

Proposition 3.31. *Let (X, Y) be a pair of random variables jointly distributed on a finite set $\Omega \times \Gamma$. Then*[13] $\mathbb{E}_Y[\mathsf{dist}(X|Y, X)] = \mathbb{E}_X[\mathsf{dist}(Y|X, Y)]$.

Proof. For $x \in \Omega$ and $y \in \Gamma$, we will use shorthands p_x, p_y, p_{xy} to denote $\Pr[X = x], \Pr[Y = y], \Pr[X = x, Y = y]$, respectively. Then we have

$$\begin{aligned}
\mathbb{E}_Y[\mathsf{dist}(X|Y, X)] &= \sum_{y\in\Gamma} p_y \mathsf{dist}(X|(Y = y), X) = \frac{1}{2}\sum_{y\in\Gamma} p_y \sum_{x\in\Omega} |p_{xy}/p_y - p_x| \\
&= \frac{1}{2}\sum_{y\in\Gamma}\sum_{x\in\Omega} |p_{xy} - p_x p_y| = \frac{1}{2}\sum_{x\in\Omega} p_x \sum_{y\in\Gamma} |p_{xy}/p_x - p_y| \\
&= \sum_{x\in\Omega} p_x \mathsf{dist}(Y|(X = x), Y) = \mathbb{E}_X[\mathsf{dist}(Y|X, Y)].
\end{aligned}$$

□

Proposition 3.32. *Let Ω be a finite set that is partitioned into subsets $S_1, \ldots, S_k$ and suppose that $\mathcal{X}$ is a distribution on Ω that is γ-close to uniform. Denote by p_i, $i = 1, \ldots k$, the probability assigned to the event S_i by*

[13] Here we are abusing the notation and denote by Y the marginal distribution of the random variable Y, and by $Y|(X = a)$ the distribution of the random variable Y conditioned on the event $X = a$.

$\mathcal{X}$. *Then*

$$\sum_{i\in[k]} p_i \cdot \mathsf{dist}(\mathcal{X}|S_i,\mathcal{U}_{S_i}) \leq 2\gamma.$$

Proof. Let $N := |\Omega|$, and define for each i, $\gamma_i := \sum_{s\in S_i} \left|\Pr_{\mathcal{X}}(s) - \frac{1}{N}\right|$, so that $\gamma_1 + \cdots + \gamma_k \leq 2\gamma$. Observe that by triangle's inequality, for every i we must have $|p_i - |S_i|/N| \leq \gamma_i$. To conclude the claim, it is enough to show that for every i, we have $\mathsf{dist}(\mathcal{X}|S_i,\mathcal{U}_{S_i}) \leq \gamma_i/p_i$. This is shown in the following.

$$\begin{aligned}
p_i \cdot \mathsf{dist}(\mathcal{X}|S_i,\mathcal{U}_{S_i}) &= \frac{p_i}{2}\sum_{s\in S_i}\left|\frac{\Pr_{\mathcal{X}}(s)}{p_i} - \frac{1}{|S_i|}\right| \\
&= \frac{1}{2}\sum_{s\in S_i}\left|\Pr_{\mathcal{X}}(s) - \frac{p_i}{|S_i|}\right| \\
&= \frac{1}{2}\sum_{s\in S_i}\left|\left(\Pr_{\mathcal{X}}(s) - \frac{1}{N}\right) + \frac{1}{|S_i|}\left(\frac{|S_i|}{N} - p_i\right)\right| \\
&\leq \frac{1}{2}\sum_{s\in S_i}\left|\Pr_{\mathcal{X}}(s) - \frac{1}{N}\right| + \frac{1}{2|S_i|}\sum_{s\in S_i}\left|\frac{|S_i|}{N} - p_i\right| \\
&\leq \frac{\gamma_i}{2} + \frac{1}{2|S_i|}\cdot|S_i|\gamma_i = \gamma_i.
\end{aligned}$$

□

The following proposition shows that any function maps close distributions to close distributions:

Proposition 3.33. *Let Ω and Γ be finite sets and f be a function from Ω to Γ. Suppose that $\mathcal{X}$ and $\mathcal{Y}$ are probability distributions on Ω and Γ, respectively, and let $\mathcal{X}'$ be a probability distribution on Ω which is δ-close to $\mathcal{X}$. Then if $f(\mathcal{X}) \sim_\epsilon \mathcal{Y}$, then $f(\mathcal{X}') \sim_{\epsilon+\delta} \mathcal{Y}$.*

Proof. Let X, X' and Y be random variables distributed according to $\mathcal{X}$, $\mathcal{X}'$, and $\mathcal{Y}$, respectively. We want to upperbound

$$\left|\Pr[f(X') \in T] - \Pr[Y \in T]\right|$$

for every $T \subseteq \Gamma$. By the triangle inequality, this is no more than

$$\left|\Pr[f(X') \in T] - \Pr[f(X) \in T]\right| + \left|\Pr[f(X) \in T] - \Pr[Y \in T]\right|.$$

Here the summand on the right hand side is upperbounded by the distance of $f(\mathcal{X})$ and $\mathcal{Y}$, that is assumed to be at most ϵ. Let $T' := \{x \in \Omega \mid f(x) \in T\}$. Then the summand on the left can be written as

$$\left|\Pr[X' \in T'] - \Pr[X \in T']\right|$$

which is at most δ by the assumption that $\mathcal{X} \sim_\delta \mathcal{X}'$. □

Omitted Details of the Proof of Corollary 3.19

Here we prove Corollary 3.19 for the case $c > 1$. The construction is similar to the case $c = 1$, and in particular the choice of m and k will remain the same. However, a subtle complication is that the expander family may not have a graph with d^m vertices and we need to adapt the extractor of Theorem 3.18 to support our parameters, still with exponentially small error. To do so, we pick a graph G in the family with N vertices, such that

$$c^{\eta m} d^m \leq N \leq c^{\eta m+1} d^m,$$

for a small absolute constant $\eta > 0$ that we are free to choose. The assumption on the expander family guarantees that such a graph exists. Let m' be the smallest integer such that $d^{m'} \geq c^{\eta m} N$. Index the vertices of G by integers in $[N]$. Note that m' will be larger than m by a constant multiplicative factor that approaches 1 as $\eta \to 0$.

For positive integers q and $p \leq q$, define the function $\mathsf{Mod}_{q,p} \colon [q] \to [p]$ by

$$\mathsf{Mod}_{q,p}(x) := 1 + (x \bmod p).$$

The extractor SFExt interprets the first m' symbols of the input as an integer u, $0 \leq u < d^{m'}$ and performs a walk on G starting from the vertex $\mathsf{Mod}_{d^{m'},N}(u+1)$, the walk being defined by the remaining input symbols. If the walk reaches a vertex v at the end, the extractor outputs $\mathsf{Mod}_{N,d^m}(v)-1$, encoded as a d-ary string of length m. A similar argument as in Theorem 3.18 can show that with our choice of the parameters, the extractor has an exponentially small error, where the error exponent is now inferior to that of Theorem 3.18 by $O(m)$, but the constant behind $O(\cdot)$ can be made arbitrarily small by choosing a sufficiently small η.

The real difficulty lies with the inverter because Mod is not a balanced function (that is, all images do not have the same number of preimages), thus we will not be able to obtain a perfect inverter. Nevertheless, it is possible to construct an inverter with a close-to-uniform output in ℓ_∞ norm. This turns out to be as good as having a perfect inverter, and thanks to the following lemma, we will still be able to use it to construct a wiretap protocol with zero leakage:

Lemma 3.34. *Suppose that $f \colon [d]^n \to [d]^m$ is a $(k, 2^{-\Omega(m)})_d$ symbol-fixing extractor and that $\mathcal{X}$ is a distribution on $[d]^n$ such that $\|\mathcal{X} - \mathcal{U}_{[d]^n}\|_\infty \leq 2^{-\Omega(m)}/d^n$. Denote by $\mathcal{X}'$ the distribution $\mathcal{X}$ conditioned on any fixing of at most $n-k$ coordinates. Then $f(\mathcal{X}') \sim_{2^{-\Omega(m)}} \mathcal{U}_{[d]^m}$.*

Proof. By Proposition 3.33, it suffices to show that $\mathcal{X}'$ is $2^{-\Omega(m)}$-close to an $(n,k)_d$ symbol-fixing source. Let $S \subseteq [d]^m$ denote the support of $\mathcal{X}'$, and let ϵ/d^n be the ℓ_∞ distance between $\mathcal{X}$ and $\mathcal{U}_{[d]^n}$, so that by our assumption, $\epsilon = 2^{-\Omega(m)}$. By the bound on the ℓ_∞ distance, we know that $\Pr_{\mathcal{X}}(S)$ is

between $\frac{|S|}{d^n}(1-\epsilon)$ and $\frac{|S|}{d^n}(1+\epsilon)$. Hence for any $x \in S$, $\Pr_{\mathcal{X}'}(x)$, which is $\Pr_{\mathcal{X}}(x)/\Pr_{\mathcal{X}}(S)$, is between $\frac{1}{|S|}\cdot\frac{1-\epsilon}{1+\epsilon}$ and $\frac{1}{|S|}\cdot\frac{1+\epsilon}{1-\epsilon}$. This differs from $1/|S|$ by at most $O(\epsilon)/|S|$. Hence, $\mathcal{X}'$ is $2^{-\Omega(m)}$-close to $\mathcal{U}_S$. □

In order to invert our new construction, we will need to construct an inverter $\mathsf{Inv}_{q,p}$ for the function $\mathsf{Mod}_{q,p}$. For that, given $x \in [p]$ we will just sample uniformly in its preimages. This is where the non-balancedness of Mod causes problems, since if p does not divide q the distribution $\mathsf{Inv}_{q,p}(\mathcal{U}_{[p]})$ is not uniform on $[q]$.

Lemma 3.35. *Suppose that $q > p$. Given a distribution $\mathcal{X}$ on $[p]$ such that $\|\mathcal{X} - \mathcal{U}_{[p]}\|_\infty \le \frac{\epsilon}{p}$, we have $\|\mathsf{Inv}_{q,p}(\mathcal{X}) - \mathcal{U}_{[q]}\|_\infty \le \frac{1}{q}\cdot\frac{p+\epsilon q}{q-p}$.*

Proof. Let $X \sim \mathcal{X}$ and $Y \sim \mathsf{Inv}_{q,p}(\mathcal{X})$. Since we invert the modulo function by taking for a given output a random preimage uniformly, $\Pr[Y = y]$ is equal to $\Pr[X = \mathsf{Mod}_{q,p}(y)]$ divided by the number of y with the same value for $\mathsf{Mod}_{q,p}(y)$. The latter number is either $\lfloor q/p \rfloor$ or $\lceil q/p \rceil$, so

$$\frac{1-\epsilon}{p\lceil q/p\rceil} \le \Pr(Y = y) \le \frac{1+\epsilon}{p\lfloor q/p\rfloor}$$

Bounding the floor and ceiling functions by $q/p \pm 1$, we obtain

$$\frac{1-\epsilon}{q+p} \le \Pr(Y = y) \le \frac{1+\epsilon}{q-p}$$

That is

$$\frac{-p-\epsilon q}{q(q+p)} \le \Pr(Y = y) - \frac{1}{q} \le \frac{p+\epsilon q}{q(q-p)}\ ,$$

which concludes the proof since this is true for all y. □

Now we describe the inverter $\mathsf{Inv}(x)$ for the extractor, again abusing the notation. First the inverter calls $\mathsf{Inv}_{N,d^m}(x)$ to obtain $x_1 \in [N]$. Then it performs a random walk on the graph, starting from x_1, to reach a vertex x_2 at the end which is inverted to obtain $x_3 = \mathsf{Inv}_{d^{m'},N}(x_2)$ as a d-ary string of length m'. Finally, the inverter outputs $y = (x_3, w)$, where w corresponds the *inverse* of the random walk of length $n - m'$. It is obvious that this procedure yields a valid preimage of x.

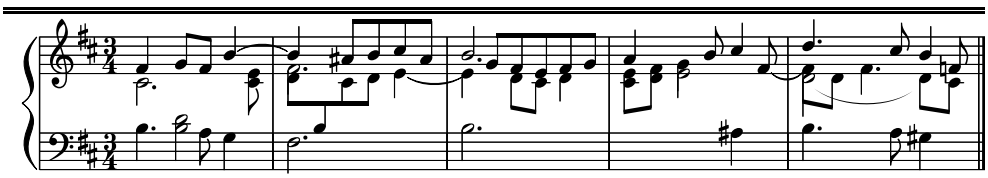
Using the previous lemma, if x is chosen uniformly, x_1 will be at ℓ_∞-distance

$$\epsilon_1 := \frac{1}{N}\cdot\frac{d^m}{N-d^m} = \frac{1}{N}O(c^{-\eta m}).$$

For a given walk, the distribution of x_2 will just be a permutation of the distribution of x_1 and applying the lemma again, we see that the ℓ_∞-distance of x_3 from the uniform distribution is

$$\epsilon_2 := \frac{1}{d^{m'}}\cdot\frac{N+\epsilon_1 d^{m'}}{d^{m'}-N} = \frac{1}{d^{m'}}O(c^{-\eta m}).$$

This is true for all the $d^{n-m'}$ possible walks so the ℓ_∞-distance of the distribution of y from uniform is bounded by $\frac{1}{d^n}O(c^{-\eta m})$. Applying Lemma 3.34 in an argument similar to Lemma 3.15 concludes the proof.



Domenico Scarlatti (1685–1757): Keyboard Sonata in B minor K. 87 (L. 33).

"War does not determine who is right—only who is left."
— Bertrand Russell

Chapter 4

Group Testing

The history of group testing is believed to date back to the second World War. During the war, millions of blood samples taken from draftees had to be subjected to a certain test, and be analyzed in order to identify a few thousand cases of syphilis. The tests were identical for all the samples. Here the idea of group testing came to a statistician called Robert Dorfman (and perhaps, a few other researchers working together with him, among them David Rosenblatt). He made a very intuitive observation, that, the samples are constantly subjected to the same test, which is extremely sensitive and remains reliable even if the sample is diluted. Therefore, it makes sense to, instead of analyzing each sample individually, pool every few samples in a group, and apply the test on the mixture of the samples. If the test outcome is negative, we will be sure that none of the samples participating in the pool are positive. On the other hand, if the outcome is positive, we know that one or more of the samples are positive, and will have to proceed with more refined, or individual, tests in order to identify the individual positives within the group.

Since the number of positives in the entire population was suspected to be in order of a few thousands—a small fraction of the population—Dorfman's idea would save a great deal of time and resources. Whether or not the idea had been eventually implemented at the time, Dorfman went on to publish a paper on the topic [49], which triggered an extensive line of research in combinatorics known today as *combinatorial group testing*.

The main challenge in group testing is to design the pools in such a way to minimize the number of tests required in order to identify the exact set of positives. Larger groups would save a lot of tests if their outcome is negative, and are rather wasteful otherwise (since in the latter case they convey a relatively small amount of information).

Of course the applications of group testing are not limited to blood sam-

pling. To mention another early example, consider a production line of electric items such as light bulbs (or resistors, capacitors, etc). As a part of the quality assurance, defective items have to be identified and discarded. Group testing can be used to aid this process. Suppose that a group of light bulbs are connected in series, and an electric current is passed through the circuit. If all the bulbs are illuminated, we can be sure than none is defective, and otherwise, we know that at least one is defective.

Since its emergence decades ago, group testing has found a large number of surprising applications that are too numerous to be extensively treated here. We particularly refer to applications in molecular biology and DNA library screening (cf. [18, 58, 102, 113, 130, 163, 164] and the references therein), multiaccess communication [162], data compression [81], pattern matching [37], streaming algorithms [38], software testing [14], compressed sensing [39], and secure key distribution [26], among others. Moreover, entire books are specifically targeted to combinatorial group testing [50, 51].

In formal terms, the classical group testing problem can be described as follows. Suppose that we wish to "learn" a Boolean vector of length n, namely $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ using as few questions as possible. Each question can ask for a single bit x_i, or more generally, specify a group of coordinates $\mathcal{I} \subseteq [n]$ ($\mathcal{I} \neq \emptyset$) and ask for the bit-wise "or" of the entries at the specified coordinates; i.e., $\bigvee_{i \in \mathcal{I}} x_i$. We will refer to this type of questions as *disjunctive queries*. Obviously, in order to be able to uniquely identify x, there is in general no better way than asking for individual bits $x_1, \ldots, x_n$ (and thus, n questions), since the number of Boolean vectors of length n is 2^n and thus, information theoretically, n bits of information is required to describe an arbitrary n-bit vector. Therefore, without imposing further restrictions on the possible realizations of the unknown vector, the problem becomes trivial.

Motivated by the blood sampling application that we just described, natural restriction that is always assumed in group testing on the unknown vector x is that it is *sparse*. Namely, for an integer parameter $d > 0$, we will assume that the number of nonzero entries of x is at most d. We will refer to such a vector as d-sparse. The number of d-sparse Boolean vectors is

$$\sum_{i=0}^{n} \binom{n}{i} = 2^{\Theta(d \log(n/d))},$$

and therefore, in principle, any d-sparse Boolean vector can be described using only $O(d \log(n/d))$ bits of information, a number that can be substantially smaller than n if $d \ll n$. The precise interpretation of the assumption "$d \ll n$" varies from a setting to another. For a substantial part of this chapter, one can think of $d = O(\sqrt{n})$. The important question in group testing that we will address in this chapter is that, whether the information-theoretic limit $\Omega(d \log(n/d))$ on the number of questions can be achieved using disjunctive queries as well.

Notation for this chapter: In this chapter we will be constantly working with Boolean vectors and their support. The *support* of a vector $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, denoted by $\mathsf{supp}(x)$, is a subset of $[n]$ such that $i \in \mathsf{supp}(x)$ if and only if $x_i = 1$. Thus the Hamming weight of x, that we will denote by $\mathsf{wgt}(x)$ can be defined as $\mathsf{wgt}(x) = |\mathsf{supp}(x)|$, and a d-sparse vector has the property that $\mathsf{wgt}(x) \leq d$.

For a matrix M, we denote by $M[i,j]$ the entry of M at the ith row and jth column. Moreover, we denote the ith entry of a vector x by $x(i)$ (assuming a one-to-one correspondence between the coordinate positions of x and natural numbers). For an $m \times n$ Boolean matrix M and $S \subseteq [n]$, we denote by $M|_S$ the $m \times |S|$ submatrix of M formed by restricting M to the columns picked by S.

For non-negative integers e_0 and e_1, we say that an ordered pair of binary vectors (x, y), each in $\{0,1\}^n$, are (e_0, e_1)-close (or x is (e_0, e_1)-close to y) if y can be obtained from x by flipping at most e_0 bits from 0 to 1 and at most e_1 bits from 1 to 0. Hence, such x and y will be $(e_0 + e_1)$-close in Hamming-distance. Further, (x, y) are called (e_0, e_1)-far if they are not (e_0, e_1)-close. Note that if x and y are seen as characteristic vectors of subsets X and Y of $[n]$, respectively, they are $(|Y \setminus X|, |X \setminus Y|)$-close. Furthermore, (x, y) are (e_0, e_1)-close if and only if (y, x) are (e_1, e_0)-close.

4.1 Measurement Designs and Disjunct Matrices

Suppose that we wish to correctly identify a d-sparse vector $x \in \{0,1\}^n$ using a reasonable amount of disjunctive queries (that we will simply refer to as "measurements"). In order to do so, consider first the following simple scheme:

1. If $n \leq 2d$, trivially measure the vector by querying $x_1, \ldots, x_n$ individually.

2. Otherwise, partition the coordinates of x into $\lfloor 2d \rfloor$ blocks of length either $\lfloor n/(2d) \rfloor$ or $\lceil n/(2d) \rceil$ each, and query the bitwise "or" of the positions within each block.

3. At least half of the measurement outcomes must be negative, since the vector x is d-sparse. Recursively run the measurements over the union of those blocks that have returned positive.

In the above procedure, each recursive call reduces the length of the vector to half or less, which implies that the depth of the recursion is $\log(n/2d)$. Moreover, since $2d$ measurements are made at each level, altogether we will have $O(d \log(n/d))$ measurements. Therefore, the simple scheme above is optimal in the sense that it attains the information-theoretic limit $\Omega(d \log(n/d))$ on the number of measurements, up to constant factors.

The main problem with this scheme is that, the measurements are *adaptive* in nature. That is, the choice of the coordinate positions defining each measurement may depend on the outcomes of the previous measurements. However, the scheme can be seen as having $O(\log(n/d))$ adaptive *stages*. Namely, each level of the recursion consists of $2d$ queries whose choices depend on the query outcomes of the previous levels, but otherwise do not depend on the outcomes of one another and can be asked in parallel.

Besides being of theoretical interest, for certain application such as those in molecular biology, adaptive schemes can be infeasible or too costly, and the "amortized" cost per test can be substantially lowered when all queries are specified and fixed before any measurements are performed. Thus, a basic goal would be to design a measurement scheme that is fully non-adaptive so that all measurements can be performed in parallel. The trivial scheme, of course, is an example of a non-adaptive scheme that achieves n measurements. The question is that, how close can one get to the information-theoretic limit $\Omega(\log(n/d))$ using a fully non-adaptive scheme? In order to answer this question, we must study the *combinatorial structure* of non-adaptive group testing schemes.

Non-adaptive measurements can be conveniently thought of in a matrix form, known as the *measurement matrix*, that is simply the incidence matrix of the set of queries. Each query can be represented by a Boolean row vector of length n that is the characteristic vector of the set of indices that participate in the query. In particular, for a query that takes a subset $\mathcal{I} \subseteq [n]$ of the coordinate positions, the corresponding vector representation would the Boolean vector of length n that is supported on the positions picked by $\mathcal{I}$. Then the measurement matrix is obtained by arranging the vector encodings on the individual queries as its rows. In particular, the measurement matrix corresponding to a set of m non-adaptive queries will be the $m \times n$ Boolean matrix that has a 1 at each position (i, j) if and only if the jth coordinate participates in the ith query. Under this notation, the measurement outcomes corresponding to a Boolean vector $x \in \{0,1\}^n$ and an $m \times n$ measurement matrix M is nothing but the Boolean vector of length m that is equal to the bit-wise "or" of those columns of M picked by the support of x. We will denote the vector of measurement outcomes by $M[x]$. For example, for the measurement matrix

$$M := \begin{pmatrix} \mathbf{0} & \mathbf{0} & 1 & \mathbf{1} & 0 & 1 & 1 & 0 \\ \mathbf{1} & \mathbf{0} & 1 & \mathbf{0} & 0 & 1 & 0 & 1 \\ \mathbf{0} & \mathbf{1} & 0 & \mathbf{1} & 0 & 1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & 1 & 0 & 1 & 1 \\ \mathbf{1} & \mathbf{0} & 1 & \mathbf{0} & 1 & 1 & 1 & 0 \end{pmatrix}$$

and Boolean vector $x := (1,1,0,1,0,0,0,0)$, we have $M[x] = (1,1,1,0,1)$, which is the bit-wise "or" of the columns shown in boldface.

Now suppose that the measurement matrix M is chosen so that it can be used to distinguish between any two d-sparse vectors. In particular, for every set $S \subseteq [n]$ of indices such that $|S| \leq d-1$, d being the sparsity parameter, the $(d-1)$-sparse vector $x \in \{0,1\}^n$ supported on S must be distinguishable from the d-sparse vector $x' \in \{0,1\}^n$ supported on $S \cup \{i\}$, for any arbitrary index $i \in [n] \setminus S$. Now observe that the Boolean function "or" is *monotone*. Namely, for a Boolean vector $(a_1, \ldots, a_n) \in \{0,1\}^n$ that is monotonically less than or equal to another vector $(b_1, \ldots, b_n) \in \{0,1\}^n$ (i.e., for every $j \in [n]$, $a_j \leq b_j$), it must be that

$$\bigvee_{j \in [n]} a_j \leq \bigvee_{j \in [n]} b_j.$$

Therefore, since we have chosen x and x' so that $\mathsf{supp}(x) \subseteq \mathsf{supp}(x')$, we must have $\mathsf{supp}(M[x]) \subseteq \mathsf{supp}(M[x'])$. Since by assumption, $M[x]$ and $M[x']$ must differ in at least one position, at least one of the rows of M must have an entry 1 at the ith row but all zeros at those corresponding to the set S. This is the idea behind the classical notion of *disjunct* matrices, formally defined below (in a slightly generalized form).

Definition 4.1. For integer parameters $d, e \geq 0$ (respectively called the *sparsity* parameter and *noise tolerance*), a Boolean matrix is (d,e)-disjunct if for every choice of $d+1$ distinct columns $C_0, C_1, \ldots, C_d$ of the matrix we have

$$|\mathsf{supp}(C_0) \setminus \cup_{i=1}^{d} \mathsf{supp}(C_i)| > e.$$

A $(d,0)$-disjunct matrix is simply called d-disjunct.

In the discussion preceding the above definition we saw that the notion of $(d-1)$-disjunct matrices is *necessary* for non-adaptive group testing, in that any non-adaptive measurement scheme must correspond to a $(d-1)$-disjunct matrix. It turns out that this notion is also sufficient, and thus precisely captures the combinatorial structure needed for non-adaptive group testing.

Theorem 4.2. *Suppose that M is an $m \times n$ matrix that is (d,e)-disjunct. Then for every pair of distinct d-sparse vectors $x, x' \in \{0,1\}^n$ such that $\mathsf{supp}(x) \not\subseteq \mathsf{supp}(x')$, we have*

$$|\mathsf{supp}(M[x]) \setminus \mathsf{supp}(M[x'])| > e. \tag{4.1}$$

Conversely, if M is such that (4.1) holds for every choice of x, x' as above, then it must be $(d-1,e)$-disjunct.

Proof. For the forward direction, let $S := \mathsf{supp}(x')$ and $i \in \mathsf{supp}(x) \setminus \mathsf{supp}(x')$. Then Definition 4.1 implies that there is a set $E \subseteq [m]$ of rows of M such that $|E| > e$ and for every $j \in E$, we have $M[i,j] = 1$ and the jth row of M restricted to the columns in S (i.e., the support of x') entirely consists of zeros. Thus, the measurement outcomes for x' at positions in E must be zeros

while those measurements have a positive outcome for x (since they include at least one coordinate, namely i, on the support of x). Therefore, (4.1) holds.

For the converse, consider any set $S \subseteq [n]$ of size at most $d-1$ and $i \in [n] \setminus S$. Consider d-sparse vectors $x, x' \in \{0,1\}^n$ such that $\mathsf{supp}(x') := S$ and $\mathsf{supp}(x) := S \cup \{i\}$. By assumption, there must be a set $E \subseteq [m]$ of size larger than e such that, for every $j \in E$, we have $M[x](j) = 1$ but $M[x'](j) = 0$. This implies that on those rows of M that are picked by E, the ith entry must be one while those corresponding to S must be zeros. Therefore, M is (d, e)-disjunct. □

From the above theorem we know that the measurement outcomes corresponding to distinct d-sparse vectors differ from one another in more than e positions provided that the measurement matrix is (d, e)-disjunct. When $e > 0$, this would allow for distinguishability of sparse vectors even in presence of *noise*. Namely, even if up to $\lfloor e/2 \rfloor$ of the measurements are allowed to be incorrect, it would still possible to uniquely reconstruct the vector being measured. For this reason, we have called the parameter e the "noise tolerance".

4.1.1 Reconstruction

So far we have focused on *combinatorial* distinguishability of sparse vectors. However, for applications unique distinguishability is by itself not sufficient and it is important to have efficient "decoding" algorithms to reconstruct the vector being measured.

Fortunately, monotonicity of the "or" function substantially simplifies the decoding problem. In particular, if two Boolean vectors x, x' such that the support of x is not entirely contained in that of x' are distinguishable by a measurement matrix, adding new elements to the support of x will never make it "less disginguishable" from x'. Moreover, observe that the proof of Theorem 4.2 never uses sparsity of the vector x. Therefore we see that, (d, e)-disjunct matrices are not only able to distinguish between d-sparse vectors, but moreover, the only Boolean vector (be it sparse or not) that may reproduce the measurement outcomes resulting from a d-sparse vector $x \in \{0,1\}^n$ is x itself. Thus, given a vector of measurement outcomes, in order to reconstruct the sparse vector being measured it suffices to produce *any* vector that is consistent with the measurement outcomes. This observation leads us to the following simple decoding algorithm, that we will call the *distance decoder*:

1. Given a measurement outcome $\tilde{y} \in \{0,1\}^m$, identify the set $S_{\tilde{y}} \subseteq [n]$ of the column indices of the measurement matrix M such that each $i \in [n]$ is in $S_{\tilde{y}}$ if and only if the ith column of M, denoted by c_i, satisfies

$$|\mathsf{supp}(c_i) \setminus \mathsf{supp}(\tilde{y})| \leq \lfloor e/2 \rfloor.$$

2. The reconstruction outcome $\tilde{x} \in \{0,1\}^n$ is the Boolean vector supported on $S_{\tilde{y}}$.

Lemma 4.3. *Let $x \in \{0,1\}^n$ be d-sparse and $y := M[x]$, where the measurement matrix M is (d,e)-disjunct. Suppose that a measurement outcome $\tilde{y}$ that has Hamming distance at most $\lfloor e/2 \rfloor$ with y is given to the distance decoder. Then the outcome $\tilde{x}$ of the decoder is equal to x.*

Proof. Since the distance decoder allows for a "mismatch" of size up to e for the columns picked by the set $S_{\tilde{y}}$, we surely know that $\mathsf{supp}(x) \subseteq S_{\tilde{y}} = \mathsf{supp}(\tilde{x})$. Now suppose that there is an index $i \in [n]$ such that $i \in S_{\tilde{y}}$ but $i \notin \mathsf{supp}(x)$. Since M is (d,e)-disjunct, we know that for the ith column c_i we have

$$|\mathsf{supp}(c_i) \setminus \mathsf{supp}(y)| > e.$$

On the other hand, since $i \in S_{\tilde{y}}$, it must be that

$$|\mathsf{supp}(c_i) \setminus \mathsf{supp}(\tilde{y})| \leq \lfloor e/2 \rfloor,$$

and moreover, by assumption we have that

$$|\mathsf{supp}(\tilde{y}) \setminus \mathsf{supp}(y)| \leq \lfloor e/2 \rfloor.$$

This is a contradiction. Therefore we must have $S_{\tilde{y}} \subseteq \mathsf{supp}(x)$, implying that $x = \tilde{x}$. □

4.1.2 Bounds on Disjunct Matrices

So far we have seen that the notion of disjunct matrices is all we need for non-adaptive group testing. But how small can the number of rows of such matrices be? Equivalently, what is the smallest number of measurements required by a non-adaptive group testing scheme that can correctly identify the support of d-sparse vectors?

4.1.2.1 Upper and Lower Bounds

In the following, we use the probabilistic method to show that, a randomly constructed matrix is with overwhelming probability disjunct, and thus obtain an upperbound on the number of the rows of disjunct matrices.

Theorem 4.4. *Let $p \in [0,1)$ be an arbitrary real parameter, and d, n be integer parameters such that $d < n$. Consider a random $m \times n$ Boolean matrix M such that each entry of M is, independently, chosen to be 1 with probability $q := 1/d$. Then there is an $m_0 = O(d^2 \log(n/d)/(1-p)^2)$ and $e = \Omega(pm/d)$ such that M is (d,e)-disjunct with probability $1 - o(1)$ provided that $m \geq m_0$.*

Proof. Consider any set S of d columns of M, and any column outside those, say the ith column where $i \notin S$. First we upper bound the probability of a *failure* for this choice of S and i, i.e., the probability that the number of the positions at the ith column corresponding to which all the columns in S have zeros is at most e. Clearly if this event happens the (d, e)-disjunct property of M would be violated. On the other hand, if for no choice of S and i a failure happens the matrix would be indeed (d, e)-disjunct.

Now we compute the failure probability p_f for a fixed S and i. A row is *good* if at that row the ith column has a 1 but all the columns in S have zeros. For a particular row, the probability that the row is good is $q(1-q)^d$. Then failure corresponds to the event that the number of good rows is at most e. The distribution of the number of good rows is binomial with mean $\mu = q(1-q)^d m$. Choose $e := pmq(1-q)^d = \Omega(pm/d)$. By a Chernoff bound, the failure probability is at most

$$\begin{aligned} p_f &\leq \exp(-(\mu - e)^2/(2\mu)) \\ &\leq \exp(-mq(1-p)^2/6) \end{aligned}$$

where the second inequality is due to the fact that $(1-q)^d = (1-1/d)^d$ is always between $1/3$ and $1/2$.

Now if we apply a union bound over all possible choices of S and i, the probability of coming up with a bad choice of M would be at most

$$n\binom{n}{d}\exp(-mq(1-p)^2/6).$$

This probability vanishes so long as $m \geq m_0$ for some $m_0 = O(d^2 \log(n/d)/(1-p)^2)$. □

The above result shows, in particular, that d-disjunct matrices with n columns and $O(d^2 \log(n/d))$ rows exist. This is by off from the information-theoretic barrier $O(d \log(n/d))$ by a multiplicative factor $O(d)$, which raises the question, whether better disjunct matrices can be found. In the literature of group testing, combinatorial lower bounds on the number of rows of disjunct matrices exist, which show that the above upper bound is almost the best one can hope for. In particular, D'yachkov and Rykov [54] have shown that the number of rows of any d-disjunct matrices has to be $\Omega(d^2 \log_d n)$. Several other concrete lower bounds on the size of disjunct matrices is known, which are all asymptotically equivalent (e.g., [63,129]). Moreover, for a nonzero noise tolerance e, the lower bounds can be extended to $\Omega(d^2 \log_d n + ed)$.

4.1.2.2 The Fixed-Input Case

The probabilistic construction of disjunct matrices presented in Theorem 4.4 almost surely produces a disjunct matrices using $O(d^2 \log(n/d))$ measurements. Obviously, due to almost-matching lower bounds, by lowering the

number of the measurement the disjunctness property cannot be assured anymore. However, the randomized nature of the designs can be used to our benefit to show that, using merely $O(d \log n)$ measurements (almost matching the information-theoretic lower bound) it is possible (with overwhelming probability) to distinguish a "fixed" d-sparse vector from any other (not necessarily sparse) vector. More precisely we have the following result, whose proof is quite similar to that of Theorem 4.4.

Theorem 4.5. *Let $p \in [0,1)$ be an arbitrary real parameter, d, n be integer parameters such that $d < n$, and $x \in \{0,1\}^n$ be a fixed d-sparse vector. Consider a random $m \times n$ Boolean matrix M such that each entry of M is, independently, chosen to be 1 with probability $q := 1/d$. Then there is an $m_0 = O(d(\log n)/(1-p)^2)$ and $e = \Omega(pm/d)$ such that, provided that $m \geq m_0$, with probability $1 - o(1)$ the following holds: For every $y \in \{0,1\}^n$, $y \neq x$, the Hamming distance between the outcomes $M[y]$ and $M[x]$ is greater than e.*

Proof. We follow essentially the same argument as the proof of Theorem 4.4, but will need a weaker union bound at the end. Call a column i of M *good* if there are more than e rows of M at which the ith column has a 1 but those on the support of x (excluding the ith column) have zeros. Now we can follow the argument in the proof of Theorem 4.4 to show that under the conditions of the theorem, with probability $1 - o(1)$, all columns of M are good (the only difference is that, the last union bound will enumerate a set of n possibilities rather than $(n-1)\binom{n}{d}$).

Now suppose that for the particular outcome of M all columns are good, and take any $y \in \{0,1\}^n$, $y \neq x$. One of the following cases must be true, and in either case, we show that $M[x]$ and $M[y]$ are different at more than e positions:

1. There is an $i \in \mathsf{supp}(y) \setminus \mathsf{supp}(x)$: Since the ith column is good, we know that for more than e rows of M, the entry at the ith column is 1 while those at $\mathsf{supp}(x)$ are all zeros. This implies that at positions corresponding to such rows, $M[y]$ must be 1 but $M[x]$ must be zero.

2. We have $\mathsf{supp}(y) \subseteq \mathsf{supp}(x)$: In this case, take any $i \in \mathsf{supp}(x) \setminus \mathsf{supp}(y)$, and again use the fact that the ith column is good to conclude that at more than e positions the outcome $M[y]$ must be zero but $M[x]$ must be 1.

□

As a corollary, the above theorem shows that, with overwhelming probability, once we fix the outcome of the random matrix M constructed by the theorem, the matrix M will be able to distinguish between *most* d-sparse vectors even in presence of any up to $\lfloor e/2 \rfloor$ incorrect measurement outcomes.

In particular, we get an average-case result, that there is a *fixed* measurement scheme with only $O(d \log n)$ measurements using which it is possible to uniquely reconstruct a randomly chosen d-sparse vector (e.g., under the uniform distribution) with overwhelming probability over the distribution from which the sparse vector is drawn.

4.1.2.3 Sparsity of the Measurements

The probabilistic construction of Theorem 4.4 results in a rather sparse matrix, namely, one with density $q = 1/d$ that decays with the sparsity parameter d. Below we show that sparsity is a necessary condition for the probabilistic construction to work at an optimal level on the number of measurements:

Lemma 4.6. *Let M be an $m \times n$ Boolean random matrix, where $m = O(d^2 \log n)$ for an integer $d > 0$, which is constructed by setting each entry independently to 1 with probability q. Then either $q = O(\log d/d)$ or otherwise the probability that M is (d, e)-disjunct (for any $e \geq 0$) approaches to zero as n grows.*

Proof. Suppose that M is an $m \times n$ matrix that is (d, e)-disjunct. Observe that, for any integer $t \in (0, d)$, if we remove any t columns of M and all the rows on the support of those columns, the matrix must remain $(d-t, e)$-disjunct. This is because any counterexample for the modified matrix being $(d-t, e)$-disjunct can be extended to a counterexample for M being (d, e)-disjunct by adding the removed columns to its support.

Now consider any t columns of M, and denote by m_0 the number of rows of M at which the entries corresponding to the chosen columns are all zeros. The expected value of m_0 is $(1-q)^t m$. Moreover, for any constant $\delta > 0$ we have

$$\Pr[m_0 > (1+\delta)(1-q)^t m] \leq \exp(-\delta^2 (1-q)^t m/4) \tag{4.2}$$

by a Chernoff bound.

Let t_0 be the largest integer for which

$$(1+\delta)(1-q)^{t_0} m \geq \log n.$$

If $t_0 < d-1$, we let $t := 1 + t_0$ above, and this makes the right hand side of (4.2) upper bounded by $o(1)$. So with probability $1 - o(1)$, the chosen t columns of M will keep m_0 at most $(1+\delta)(1-q)^t m$, and removing those columns and m_0 rows on their union leaves the matrix $(d - t_0 - 1, e)$-disjunct, which obviously requires at least $\log n$ rows (as even a $(1, 0)$-disjunct matrix needs so many rows). Therefore, we must have

$$(1+\delta)(1-q)^t m \geq \log n$$

or otherwise (with overwhelming probability) M will not be (d, e)-disjunct. But the latter inequality is not satisfied by the assumption on t_0. So if $t_0 < d-1$, little chance remains for M to be (d, e)-disjunct.

Now consider the case $t_0 \geq d-1$. Thus, by the choice of t_0, we must have

$$(1+\delta)(1-q)^{d-1}m \geq \log n.$$

The above inequality implies that we must have

$$q \leq \frac{\log(m(1+\delta)/\log n)}{d-1},$$

which, for $m = O(d^2 \log n)$ gives $q = O(\log d/d)$. □

4.2 Noise resilient schemes and approximate reconstruction

So far, we have introduced the notion of (d, e)-disjunct matrices that can be used in non-adaptive group testing schemes to identify d-sparse vectors up to a number of measurement errors depending on the parameter e. However, as the existing lower bounds suggest, the number of rows of such matrices cannot reach to the information-theoretic optimum $O(d \log(n/d))$ and moreover, the noise tolerance e can be at most a factor $1/d$ of the number of measurements. This motivates two natural questions:

1. Can the number of measurements be lowered at the cost of causing a slight amount of "confusion"? We know, by Theorem 4.5 that, it is possible to identify sparse vectors on average using only $O(d \log n)$ measurements. But can something be said in the *worst case* ?

2. What can be said if the amount of possible errors can be substantially high; e.g., when a constant fraction of the measurements can produce false outcomes?

In order to answer the above questions, in this section we introduce a notion of measurement schemes that can be "more flexible" than that of disjunct matrices, and aims to study the trade-off between the amount of errors expected on the measurements versus the ambiguity of the reconstruction. More formally we define the following notion.

Definition 4.7. Let $m, n, d, e_0, e_1, e_0', e_1'$ be integers. An $m \times n$ measurement matrix A is called (e_0, e_1, e_0', e_1')-resilient for d-sparse vectors if, for every $y \in \{0,1\}^m$ there exists $z \in \{0,1\}^n$ (called a *valid decoding of* y) such that for every $x \in \{0,1\}^n$, whenever (x, z) are (e_0', e_1')-far, $(A[x], y)$ are (e_0, e_1)-far[1].

[1] In particular this means that for every $x, x' \in \{0,1\}^n$, if $(A[x], A[x'])$ are (e_0, e_1)-close, then x and x' must be $(e_0' + e_1', e_0' + e_1')$-close.

The matrix A is called explicit if it can be computed in polynomial time in its size, and *fully explicit* if each entry of the matrix can be computed in time $\mathsf{poly}(m, \log n)$.

Intuitively, the definition states that two measurements are allowed to be confused only if they are produced from close vectors. The parameters e_0 and e'_0 correspond to amount of tolerable *false positives* on the measurement outcomes and reconstructed vector, where by false positive we mean an error caused by mistaking a 0 for 1. Similarly, e_1 and e'_1 define the amount of tolerable *false negatives* on both sides, where a false negative occurs when a bit that actually must be 1 is flipped to 0.

In particular, an (e_0, e_1, e'_0, e'_1)-resilient matrix gives a group testing scheme that reconstructs the sparse vector up to e'_0 false positives and e'_1 false negatives even in the presence of e_0 false positives and e_1 false negatives in the measurement outcome. Under this notation, unique (exact) decoding would be possible using an $(e_0, e_1, 0, 0)$-resilient matrix if the amount of measurement errors is bounded by at most e_0 false positives and e_1 false negatives. However, when $e'_0 + e'_1$ is positive, decoding may require a bounded amount of ambiguity, namely, up to e'_0 false positives and e'_1 false negatives in the decoded sequence.

Observe that the special case of $(0, 0, 0, 0)$-resilient matrices corresponds to the classical notion of d-disjunct matrices, while a (d, e)-disjunct matrix would give a $(\lfloor e/2 \rfloor, \lfloor e/2 \rfloor, 0, 0)$-resilient matrix for d-sparse vectors.

Definition 4.7 is in fact reminiscent of *list-decoding* in error-correcting codes, but with the stronger requirement that the list of decoding possibilities must consist of vectors that are close to one another.

4.2.1 Negative Results

In coding theory, it is possible to construct codes that can tolerate up to a constant fraction of adversarially chosen errors and still guarantee unique decoding. Hence it is natural to wonder whether a similar possibility exists in group testing, namely, whether there is a measurement matrix that is robust against a constant fraction of adversarial errors and still recovers the measured vector exactly. We already have mentioned that this is in general not possible, since any (d, e)-disjunct matrix (a notion that is necessary for this task) requires at least de rows, and thus the fraction of tolerable errors by disjunct matrices cannot be above $1/d$. Below we extend this result to the more "asymmetric" notion of resilient matrices, and show that the fraction of tolerable false positives and false negatives must be both below $1/d$.

Lemma 4.8. *Suppose that an $m \times n$ measurement matrix M is (e_0, e_1, e'_0, e'_1)-resilient for d-sparse vectors. Then $(\max\{e_0, e_1\} + 1)/(e'_0 + e'_1 + 1) \leq m/d$.*

Proof. We use similar arguments as those used in [20, 75] in the context of black-box hardness amplification in NP: Define a partial ordering $\prec$ between binary vectors using bit-wise comparisons (with $0 < 1$). Let $t := d/(e'_0+e'_1+1)$ be an integer[2], and consider any monotonically increasing sequence of vectors $x_0 \prec \cdots \prec x_t$ in $\{0,1\}^n$ where x_i has weight $i(e'_0 + e'_1 + 1)$. Thus, x_0 and x_t will have weights zero and d, respectively. Note that we must also have $M[x_0] \prec \cdots \prec M[x_t]$ due to monotonicity of the "or" function.

A fact that is directly deduced from Definition 4.7 is that, for every $x, x' \in \{0,1\}^n$, if $(M[x], M[x'])$ are (e_0, e_1)-close, then x and x' must be $(e'_0 + e'_1, e'_0 + e'_1)$-close. This can be seen by setting $y := M[x']$ in the definition, for which there exists a valid decoding $z \in \{0,1\}^n$. As $(M[x], y)$ are (e_0, e_1)-close, the definition implies that (x, z) must be (e'_0, e'_1)-close. Moreover, $(M[x'], y)$ are $(0, 0)$-close and thus, (e_0, e_1)-close, which implies that (z, x') must be (e'_1, e'_0)-close. Thus by the triangle inequality, (x, x') must be $(e'_0 + e'_1, e'_0 + e'_1)$-close.

Now, observe that for all i, (x_i, x_{i+1}) are $(e'_0 + e'_1, e'_0 + e'_1)$-far, and hence, their encodings must be (e_0, e_1)-far, by the fact we just mentioned. In particular this implies that $M[x_t]$ must have weight at least $t(e_0+1)$, which must be trivially upper bounded by m. Hence it follows that $(e_0 + 1)/(e'_0 + e'_1 + 1) \leq m/d$. Similarly we can also show that $(e_1 + 1)/(e'_0 + e'_1 + 1) \leq m/d$. □

As shown by the lemma above, tolerance of a measurement matrix against a constant fraction of errors would make an ambiguity of order $\Omega(d)$ in the decoding inevitable, irrespective of the number of measurements. For most applications this might be an unsatisfactory situation, as even a close estimate of the set of positives might not reveal whether any particular individual is defective or not, and in certain scenarios (such as an epidemic disease or industrial quality assurance) it is unacceptable to miss any defective individuals. This motivates us to focus on approximate reconstructions with *one-sided* error. Namely, we will require the support of the reconstruction $\hat{x}$ to always contain the support of the original vector x being measured, and be possibly larger by up to $O(d)$ positions. It can be argued that, for most applications, such a scheme is as good as exact reconstruction, as it allows one to significantly narrow-down the set of defectives to up to $O(d)$ *candidate positives*. In particular, as observed in [93], one can use a *second stage* if necessary and individually test the resulting set of candidates, using more reliable measurements, to identify the exact set of positives. In the literature, such schemes are known as *trivial two-stage* schemes.

The trade-off given by the following lemma only focuses on false negatives and is thus useful for trivial two-stage schemes:

[2]For the sake of simplicity in this presentation we ignore the fact that certain fractions might in general give non-integer values. However, it should be clear that this will cause no loss of generality.

Lemma 4.9. *Suppose that an $m \times n$ measurement matrix M is (e_0, e_1, e_0', e_1')-resilient for d-sparse vectors. Then for every $\epsilon > 0$, either*

$$e_1 < \frac{(e_1' + 1)m}{\epsilon d}$$

or

$$e_0' \geq \frac{(1-\epsilon)(n-d+1)}{(e_1'+1)^2}.$$

Proof. Let $x \in \{0,1\}^n$ be chosen uniformly at random among vectors of weight d. Randomly flip $e_1' + 1$ of the bits on the support of x to 0, and denote the resulting vector by x'. Using the partial ordering $\prec$ in the proof of the last lemma, it is obvious that $x' \prec x$, and hence, $M[x'] \prec M[x]$. Let b denote any disjunction of a number of coordinates in x and b' the same disjunction in x'. We must have

$$\Pr[b' = 0 | b = 1] \leq \frac{e_1' + 1}{d},$$

as for b to be 1 at least one of the variables on the support of x must be present in the disjunction and one particular such variable must necessarily be flipped to bring the value of b' down to zero. Using this, the expected Hamming distance between $M[x]$ and $M[x']$ can be bounded as follows:

$$\mathbb{E}[\mathsf{dist}(M[x], M[x'])] = \sum_{i \in [m]} \mathbb{1}(M[x]_i = 1 \wedge M[x']_i = 0) \leq \frac{e_1' + 1}{d} \cdot m,$$

where the expectation is over the randomness of x and the bit flips, $\mathsf{dist}(\cdot,\cdot)$ denotes the Hamming distance between two vectors, and $\mathbb{1}(\cdot)$ denotes an indicator predicate.

Fix a particular choice of x' that keeps the expectation at most $(e_1'+1)m/d$. Now the randomness is over the possibilities of x, that is, flipping up to $e_1'+1$ zero coordinates of x' randomly. Denote by $\mathcal{X}$ the set of possibilities of x for which $M[x]$ and $M[x']$ are $\frac{(e_1'+1)m}{\epsilon d}$-close, and by $\mathcal{S}$ the set of all vectors that are monotonically larger than x' and are $(e_1'+1)$-close to it. Obviously, $\mathcal{X} \subseteq \mathcal{S}$, and, by Markov's inequality, we know that $|\mathcal{X}| \geq (1-\epsilon)|\mathcal{S}|$.

Let z be any valid decoding of $M[x']$, Thus, (x', z) must be (e_0', e_1')-close. Now assume that $e_1 \geq \frac{(e_1'+1)m}{\epsilon d}$ and consider any $x \in \mathcal{X}$. Hence, $(M[x], M[x'])$ are (e_0, e_1)-close and (x, z) must be (e_0', e_1')-close by Definition 4.7. Regard x, x', z as the characteristic vectors of sets $X, X', Z \subseteq [n]$, respectively, where $X' \subseteq X$. We know that $|X \setminus Z| \leq e_1'$ and $|X \setminus X'| = e_1' + 1$. Therefore,

$$|(X \setminus X') \cap Z| = |X \setminus X'| - |X \setminus Z| + |X' \setminus Z| > 0, \tag{4.3}$$

and z must take at least one nonzero coordinate from $\mathsf{supp}(x) \setminus \mathsf{supp}(x')$.

Now we construct an $(e_1'+1)$-hypergraph[3] H as follows: The vertex set is $[n] \setminus \mathsf{supp}(x')$, and for every $x \in \mathcal{X}$, we put a hyperedge containing $\mathsf{supp}(x) \setminus$

[3] See Appendix 4.A for definitions.

$\mathsf{supp}(x')$. The density of this hypergraph is at least $1-\epsilon$, by the fact that $|\mathcal{X}| \geq (1-\epsilon)\mathcal{S}$. Now Lemma 4.34 implies that H has a matching of size at least

$$t := \frac{(1-\epsilon)(n-d+1)}{(e_1'+1)^2}.$$

As by (4.3), $\mathsf{supp}(z)$ must contain at least one element from the vertices in each hyperedge of this matching, we conclude that $|\mathsf{supp}(z) \setminus \mathsf{supp}(x')| \geq t$, and that $e_0' \geq t$. □

The lemma above shows that if one is willing to keep the number e_1' of false negatives in the reconstruction at the zero level (or bounded by a constant), only an up to $O(1/d)$ fraction of false negatives in the measurements can be tolerated (regardless of the number of measurements), unless the number e_0' of false positives in the reconstruction grows to an enormous amount (namely, $\Omega(n)$ when $n-d = \Omega(n)$) which is certainly undesirable.

Recall that exact reconstruction of d-sparse vectors of length n, even in a noise-free setting, requires at least $\Omega(d^2 \log_d n)$ non-adaptive measurements. However, it turns out that there is no such restriction when an approximate reconstruction is sought for, except for the following bound which can be shown using simple counting and holds for adaptive noiseless schemes as well:

Lemma 4.10. *Let M be an $m \times n$ measurement matrix that is $(0, 0, e_0', e_1')$-resilient for d-sparse vectors. Then*

$$m \geq d\log(n/d) - d - e_0' - O(e_1' \log((n-d-e_0')/e_1')),$$

where the last term is defined to be zero for $e_1' = 0$.

Proof. The proof is a simple counting argument. For integers $a > b > 0$, we use the notation $V(a,b)$ for the volume of a Hamming ball of radius b in $\{0,1\}^a$. It is given by

$$V(a,b) = \sum_{i=0}^{b} \binom{a}{i} \leq 2^{ah(b/a)},$$

where $h(\cdot)$ is the binary entropy function defined as

$$h(x) := -x\log_2(x) - (1-x)\log_2(1-x),$$

and thus

$$\log V(a,b) \leq b\log\frac{a}{b} + (a-b)\log\frac{a}{a-b} = \Theta(b\log(a/b)).$$

Also, denote by $V'(a,b,e_0,e_1)$ the number of vectors in $\{0,1\}^a$ that are (e_0,e_1)-close to a fixed b-sparse vector. Obviously, $V'(a,b,e_0,e_1) \leq V(b,e_0)V(a-$

b, e_1). Now consider any (without loss of generality, deterministic) reconstruction algorithm D and let X denote the set of all vectors in $\{0,1\}^n$ that it returns for some noiseless encoding; that is,

$$X := \{x \in \{0,1\}^n \mid \exists y \in \mathcal{B}, x = D(A[y])\},$$

where $\mathcal{B}$ is the set of d-sparse vectors in $\{0,1\}^n$. Notice that all vectors in X must be $(d+e'_0)$-sparse, as they have to be close to the corresponding "correct" decoding. For each vector $x \in X$ and $y \in \mathcal{B}$, we say that x is *matching* to y if (y, x) are (e'_0, e'_1)-close. A vector $x \in X$ can be matching to at most $v := V'(n, d + e'_0, e'_0, e'_1)$ vectors in $\mathcal{B}$, and we upper bound $\log v$ as follows:

$$\log v \leq \log V(n-d-e'_0, e'_1) + \log V(d+e'_0, e'_0) = O(e'_1 \log((n-d-e'_0)/e'_1)) + d + e'_0,$$

where the term inside $O(\cdot)$ is interpreted as zero when $e'_1 = 0$. Moreover, every $y \in \mathcal{B}$ must have at least one matching vector in X, namely, $D(M[y])$. This means that $|X| \geq |\mathcal{B}|/v$, and that

$$\log |X| \geq \log |\mathcal{B}| - \log v \geq d \log(n/d) - d - e'_0 - O(e'_1 \log((n - d - e'_0)/e'_1)).$$

Finally, we observe that the number of measurements has to be at least $|X|$ to enable D to output all the vectors in X. □

According to the lemma, even in the noiseless scenario, any reconstruction method that returns an approximation of the sparse vector up to $e'_0 = O(d)$ false positives and without false negatives will require $\Omega(d \log(n/d))$ measurements. As we will show in the next section, an upper bound of $O(d \log n)$ is in fact attainable even in a highly noisy setting using only non-adaptive measurements. This in particular implies an asymptotically optimal trivial two-stage group testing scheme.

4.2.2 A Noise-Resilient Construction

In this section we introduce our general construction and design measurement matrices for testing d-sparse vectors in $\{0,1\}^n$. The matrices can be seen as adjacency matrices of certain unbalanced bipartite graphs constructed from good randomness condensers or extractors. The main technique that we use to show the desired properties is the *list-decoding view* of randomness condensers, extractors, and expanders, developed over the recent years starting from the work of Ta-Shma and Zuckerman on *extractor codes* [149] and followed by Guruswami, Umans, Vadhan [78] and Vadhan [155].

4.2.2.1 Construction from Condensers

We start by introducing the terms and tools that we will use in our construction and its analysis.

Definition 4.11. (mixtures, agreement, and agreement list) Let Σ be a finite set. A *mixture* over Σ^n is an n-tuple $S := (S_1, \ldots, S_n)$ such that every S_i, $i \in [n]$, is a nonempty subset of Σ.

The *agreement* of $w := (w_1, \ldots w_n) \in \Sigma^n$ with S, denoted by $\mathsf{Agr}(w, S)$, is the quantity

$$\frac{1}{n}|\{i \in [n] \colon w_i \in S_i\}|.$$

Moreover, we define the quantity

$$\mathsf{wgt}(S) := \sum_{i \in [n]} |S_i|$$

and

$$\rho(S) := \mathsf{wgt}(S)/(n|\Sigma|),$$

where the latter is the expected agreement of a random vector with S.

For example, consider a mixture $S := (S_1, \ldots, S_8)$ over $[4]^8$ where $S_1 := \emptyset, S_2 := \{1, 3\}, S_3 := \{1, 2\}, S_4 := \{1, 4\}, S_5 := \{1\}, S_6 := \{3\}, S_7 := \{4\}, S_8 := \{1, 2, 3, 4\}$. For this example, we have

$$\mathsf{Agr}((1, 3, 2, 3, 4, 3, 4, 4), S) = 5/8,$$

and $\rho(S) = 13/32$.

For a code $\mathcal{C} \subseteq \Sigma^n$ and $\alpha \in (0, 1]$, the *α-agreement list* of $\mathcal{C}$ with respect to S, denoted by $\mathsf{LIST}_{\mathcal{C}}(S, \alpha)$, is defined as the set[4]

$$\mathsf{LIST}_{\mathcal{C}}(S, \alpha) := \{c \in \mathcal{C} \colon \mathsf{Agr}(c, S) > \alpha\}.$$

Definition 4.12. (induced code) Let $f \colon \Gamma \times \Omega \to \Sigma$ be a function mapping a finite set $\Gamma \times \Omega$ to a finite set Σ. For $x \in \Gamma$, we use the shorthand $f(x)$ to denote the vector $y := (y_i)_{i \in \Omega}$, $y_i := f(x, i)$, whose coordinates are indexed by the elements of Ω in a fixed order. The *code induced by f*, denoted by $\mathcal{C}(f)$ is the set

$$\{f(x) \colon x \in \Gamma\}.$$

The induced code has a natural encoding function given by $x \mapsto f(x)$.

Definition 4.13. (codeword graph) Let $\mathcal{C} \subseteq \Sigma^n$, $|\Sigma| = q$, be a q-ary code. The *codeword graph* of $\mathcal{C}$ is a bipartite graph with left vertex set $\mathcal{C}$ and right vertex set $n \times \Sigma$, such that for every $x = (x_1, \ldots, x_n) \in \mathcal{C}$, there is an edge between x on the left and $(1, x_1), \ldots, (n, x_n)$ on the right. The *adjacency matrix* of the codeword graph is an $n|\Sigma| \times |\mathcal{C}|$ binary matrix whose (i, j)th entry is 1 if and only if there is an edge between the ith right vertex and the jth left vertex.

$$
\begin{array}{c}
\begin{array}{cccccccccc}
0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & & 1111 \\
= & = & = & = & = & = & = & = & & = \\
x & x & x & x & x & x & x & x & & x
\end{array} \\
\begin{pmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \dots & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \dots & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \dots & 1
\end{pmatrix}
\begin{array}{l} y=1 \\ y=2 \\ y=3 \end{array} \\
f(x,y)
\end{array}
$$

$$
\begin{array}{c}
\begin{array}{cccccccccc}
0000 & 0001 & 0010 & 0011 & 0100 & 0101 & 0110 & 0111 & & 1111 \\
= & = & = & = & = & = & = & = & & = \\
x & x & x & x & x & x & x & x & & x
\end{array} \\
\begin{array}{l} (y,f(x,y)) \\ (1,0) \\ (1,1) \\ (2,0) \\ (2,1) \\ (3,0) \\ (3,1) \end{array}
\begin{pmatrix}
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & \dots & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \dots & 1 \\
\hline
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & \dots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \dots & 1 \\
\hline
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & \dots & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \dots & 1
\end{pmatrix}
\end{array}
$$

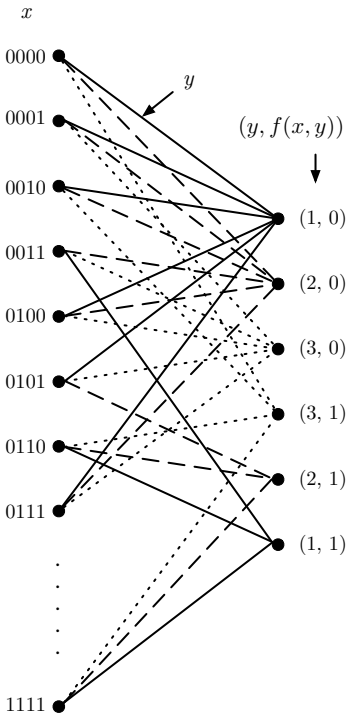


Figure 4.1: A function $f\colon \{0,1\}^4 \times [3] \to \{0,1\}$ with its truth table (top left), codeword graph of the induced code (right), and the adjacency matrix of the graph (bottom left). Solid, dashed and dotted edges in the graph respectively correspond to the choices $y = 1$, $y = 2$, and $y = 3$ of the second argument.

A simple example of a function with its truth table, codeword graph of the induced code along with its adjacency matrix is given in Figure 4.1.

The following theorem is a straightforward generalization of the result in [149] that is also shown in [78] (we have included a proof for completeness):

Theorem 4.14. *Let $f\colon \{0,1\}^{\tilde{n}} \times \{0,1\}^t \to \{0,1\}^{\tilde{\ell}}$ be a strong $k \to_\epsilon k'$ condenser, and $\mathcal{C} \subseteq \Sigma^{2^t}$ be its induced code, where $\Sigma := \{0,1\}^{\tilde{\ell}}$. Then for any mixture S over Σ^{2^t} we have*

$$|\mathsf{LIST}_{\mathcal{C}}(S, \rho(S)2^{\tilde{\ell}-k'} + \epsilon)| < 2^k.$$

Proof. Index the coordinates of S by the elements of $\{0,1\}^t$ and denote the ith coordinate by S_i. Let Y be any random variable with min-entropy at least $t + k'$ distributed on $\mathbb{F}_2^{t+k'}$. Define an information-theoretic test $T\colon \{0,1\}^{\tilde{\ell}} \times \{0,1\}^t \to \{0,1\}$ as follows: $T(x,i) = 1$ if and only if $x \in S_i$. Observe that

$$\Pr[T(Y) = 1] \leq \mathsf{wgt}(S)2^{-(t+k')} = \rho(S)2^{\tilde{\ell}-k'},$$

[4]When $\alpha = 1$, we consider codewords with full agreement with the mixture.

and that for every vector $w \in (\{0,1\}^\ell)^{2^t}$,

$$\Pr_{i \sim \mathcal{U}_t}[T(w_i, i) = 1] = \mathsf{Agr}(w, S).$$

Now, let the random variable $X = (X_1, \ldots, X_{2^t})$ be uniformly distributed on the codewords in $\mathsf{LIST}_{\mathcal{C}}(S, \rho(S)2^{\tilde{\ell}-k'} + \epsilon)$ and $Z \sim \mathcal{U}_t$. Thus, from Definition 4.11 we know that

$$\Pr_{X,Z}[T(X_Z, Z) = 1] > \rho(S)2^{\tilde{\ell}-k'} + \epsilon.$$

As the choice of Y was arbitrary, this implies that T is able to distinguish between the distribution of (Z, X) and any distribution on $\{0,1\}^{t+\tilde{\ell}}$ with min-entropy at least $t + k'$, with bias greater than ϵ, which by the definition of condensers implies that the min-entropy of X must be less than k, or

$$|\mathsf{LIST}_{\mathcal{C}}(S, \rho(S)2^{\tilde{\ell}-k'} + \epsilon)| < 2^k.$$

□

Now using the above tools, we are ready to describe and analyze our construction of error-resilient measurement matrices. We first state a general result without specifying the parameters of the condenser, and then instantiate the construction with various choices of the condenser, resulting in matrices with different properties.

Theorem 4.15. *Let $f\colon \{0,1\}^{\tilde{n}} \times \{0,1\}^t \to \{0,1\}^{\tilde{\ell}}$ be a strong $k \to_\epsilon k'$ condenser, and $\mathcal{C}$ be its induced code. Suppose that the parameters $p, \nu, \gamma > 0$ are chosen so that*

$$(p + \gamma)2^{\tilde{\ell}-k'} + \nu/\gamma < 1 - \epsilon,$$

and $d := \gamma 2^{\tilde{\ell}}$. Then the adjacency matrix of the codeword graph of $\mathcal{C}$ (which has $m := 2^{t+\tilde{\ell}}$ rows and $n := 2^{\tilde{n}}$ columns) is a $(pm, (\nu/d)m, 2^k - d, 0)$-resilient measurement matrix for d-sparse vectors. Moreover, it allows for a reconstruction algorithm with running time $O(mn)$.

Proof. Define $L := 2^{\tilde{\ell}}$ and $T := 2^t$. Let M be the adjacency matrix of the codeword graph of $\mathcal{C}$. It immediately follows from the construction that the number of rows of M (denoted by m) is equal to TL. Moreover, notice that the Hamming weight of each column of M is exactly T.

Let $x \in \{0,1\}^n$ and denote by $y \in \{0,1\}^m$ its encoding, i.e., $y := M[x]$, and by $\hat{y} \in \{0,1\}^m$ a *received word*, or a *noisy* version of y.

The encoding of x can be schematically viewed as follows: The coefficients of x are assigned to the left vertices of the codeword graph and the encoded bit on each right vertex is the bitwise "or" of the values of its neighbors.

The coordinates of x can be seen in one-to-one correspondence with the codewords of $\mathcal{C}$. Let $X \subseteq \mathcal{C}$ be the set of codewords corresponding to the

support of x. The coordinates of the noisy encoding $\hat{y}$ are indexed by the elements of $[T] \times [L]$ and thus, $\hat{y}$ naturally defines a mixture $S = (S_1, \ldots, S_T)$ over $[L]^T$, where S_i contains j iff $\hat{y}$ at position (i, j) is 1.

Observe that $\rho(S)$ is the relative Hamming weight (denoted below by $\delta(\cdot)$) of $\hat{y}$; thus, we have

$$\rho(S) = \delta(\hat{y}) \leq \delta(y) + p \leq d/L + p = \gamma + p,$$

where the last inequality comes from the fact that the relative weight of each column of M is exactly $1/L$ and that x is d-sparse.

Furthermore, from the assumption we know that the number of false negatives in the measurement is at most $\nu TL/d = \nu T/\gamma$. Therefore, any codeword in X must have agreement at least $1 - \nu/\gamma$ with S. This is because S is indeed constructed from a mixture of the elements in X, modulo false positives (that do not decrease the agreement) and at most $\nu T/\gamma$ false negatives each of which can reduce the agreement by at most $1/T$.

Accordingly, we consider a decoder which, similar to the distance decoder that we have introduced before, simply outputs a binary vector $\hat{x}$ supported on the coordinates corresponding to those codewords of $\mathcal{C}$ that have agreement larger than $1 - \nu/\gamma$ with S. Clearly, the running time of the decoder is linear in the size of the measurement matrix.

By the discussion above, $\hat{x}$ must include the support of x. Moreover, Theorem 4.14 applies for our choice of parameters, implying that $\hat{x}$ must have weight less than 2^k. □

4.2.2.2 Instantiations

Now we instantiate the general result given by Theorem 4.15 with various choices of the underlying condenser, among the results discussed in Section 2.3, and compare the obtained parameters. First, we consider two extreme cases, namely, a non-explicit optimal condenser with zero overhead (i.e., extractor) and then a non-explicit optimal condenser with zero loss (i.e., lossless condenser) and then consider how known explicit constructions can approach the obtained bounds. A summary of the obtained results is given in Table 4.1.

Optimal Extractors

Recall Radhakrishan and Ta-Shma's non-constructive bound that for every choice of the parameters $k, \tilde{n}, \epsilon$, there is a strong (k, ϵ)-extractor with input length $\tilde{n}$, seed length $t = \log(\tilde{n} - k) + 2\log(1/\epsilon) + O(1)$ and output length $\tilde{\ell} = k - 2\log(1/\epsilon) - O(1)$, and that the bound is achieved by a random function. Plugging this result in Theorem 4.15, we obtain a non-explicit measurement matrix from a simple, randomized construction that achieves the desired trade-off with high probability:

Table 4.1: A summary of constructions in Section 4.2.2. The parameters $\alpha \in [0,1)$ and $\delta \in (0,1]$ are arbitrary constants, m is the number of measurements, e_0 (resp., e_1) the number of tolerable false positives (resp., negatives) in the measurements, and e_0' is the number of false positives in the reconstruction. The fifth column shows whether the construction is explicit (Exp) or randomized (Rnd), and the last column shows the running time of the reconstruction algorithm.

m	e_0	e_1	e_0'	Exp/ Rnd	Rec. Time
$O(d \log n)$	αm	$\Omega(m/d)$	$O(d)$	Rnd	$O(mn)$
$O(d \log n)$	$\Omega(m)$	$\Omega(m/d)$	δd	Rnd	$O(mn)$
$O(d^{1+o(1)} \log n)$	αm	$\Omega(m/d)$	$O(d)$	Exp	$O(mn)$
$d \cdot \mathsf{quasipoly}(\log n)$	$\Omega(m)$	$\Omega(m/d)$	δd	Exp	$O(mn)$
$d \cdot \mathsf{quasipoly}(\log n)$	αm	$\Omega(m/d)$	$O(d)$	Exp	$\mathsf{poly}(m)$
$\mathsf{poly}(d)\mathsf{poly}(\log n)$	$\mathsf{poly}(d)\mathsf{poly}(\log n)$	$\Omega(e_0/d)$	δd	Exp	$\mathsf{poly}(m)$

Corollary 4.16. *For every choice of constants $p \in [0,1)$ and $\nu \in [0,\nu_0)$, $\nu_0 := (\sqrt{5-4p}-1)^3/8$, and positive integers d and $n \geq d$, there is an $m \times n$ measurement matrix, where $m = O(d \log n)$, that is $(pm, (\nu/d)m, O(d), 0)$-resilient for d-sparse vectors of length n and allows for a reconstruction algorithm with running time $O(mn)$.*

Proof. For simplicity we assume that $n = 2^{\tilde{n}}$ and $d = 2^{\tilde{d}}$ for positive integers $\tilde{n}$ and $\tilde{d}$. However, it should be clear that this restriction will cause no loss of generality and can be eliminated with a slight change in the constants behind the asymptotic notations.

We instantiate the parameters of Theorem 4.15 using an optimal strong extractor. If $\nu = 0$, we choose γ, ϵ small constants such that $\gamma + \epsilon < 1 - p$. Otherwise, we choose $\gamma := \sqrt[3]{\nu}$, which makes $\nu/\gamma = \sqrt[3]{\nu^2}$, and $\epsilon < 1 - p - \sqrt[3]{\nu} - \sqrt[3]{\nu^2}$. (One can easily see that the right hand side of the latter inequality is positive for $\nu < \nu_0$). Hence, the condition $p + \nu/\gamma < 1 - \epsilon - \gamma$ required by Theorem 4.15 is satisfied.

Let $r = 2\log(1/\epsilon) + O(1) = O(1)$ be the entropy loss of the extractor for error ϵ, and set up the extractor for min-entropy $k = \log d + \log(1/\gamma) + r$, which means that $K := 2^k = O(d)$ and $L := 2^{\tilde{\ell}} = d/\gamma = O(d)$. Now we can apply Theorem 4.15 and conclude that the measurement matrix is $(pm, (\nu/d)m, O(d), 0)$-resilient. The seed length required by the extractor is $t \leq \log \tilde{n} + 2\log(1/\epsilon) + O(1)$, which gives $T := 2^t = O(\log n)$. Therefore, the number of measurements will be $m = TL = O(d \log n)$. □

Optimal Lossless Condensers

Now we instantiate Theorem 4.15 with an optimal strong lossless condenser with input length $\tilde{n}$, entropy requirement k, seed length $t = \log \tilde{n} + \log(1/\epsilon) + O(1)$ and output length $\tilde{\ell} = k + \log(1/\epsilon) + O(1)$. Thus we get the following corollary.

Corollary 4.17. *For positive integers $n \geq d$ and every constant $\delta > 0$ there is an $m \times n$ measurement matrix, where $m = O(d \log n)$, that is $(\Omega(m), \Omega(1/d)m, \delta d, 0)$-resilient for d-sparse vectors of length n and allows for a reconstruction algorithm with running time $O(mn)$.*

Proof. We will use the notation of Theorem 4.15 and apply it using an optimal strong lossless condenser. This time, we set up the condenser with error $\epsilon := \frac{1}{2}\delta/(1+\delta)$ and min-entropy k such that $K := 2^k = d/(1-2\epsilon)$. As the error is a constant, the overhead and hence $2^{\tilde{\ell}-k}$ will also be a constant. The seed length is $t = \log(\tilde{n}/\epsilon) + O(1)$, which makes $T := 2^t = O(\log n)$. As $L := 2^{\tilde{\ell}} = O(d)$, the number of measurements becomes $m = TL = O(d \log n)$, as desired.

Moreover, note that our choice of K implies that $K - d = \delta d$. Thus we only need to choose p and ν appropriately to satisfy the condition

$$(p+\gamma)L/K + \nu/\gamma < 1 - \epsilon, \tag{4.4}$$

where $\gamma = d/L = K/(L(1+\delta))$ is a constant, as required by the lemma. Substituting for γ in (4.4) and after simple manipulations, we get the condition

$$pL/K + \nu(L/K)(1+\delta) < \frac{\delta}{2(1+\delta)},$$

which can be satisfied by choosing p and ν to be appropriate positive constants. □

Both results obtained in Corollaries 4.16 and 4.17 almost match the lower bound of Lemma 4.10 for the number of measurements. However, we note the following distinction between the two results: Instantiating the general construction of Theorem 4.15 with an extractor gives us a sharp control over the fraction of tolerable errors, and in particular, we can obtain a measurement matrix that is robust against *any* constant fraction (bounded from 1) of false positives. However, the number of potential false positives in the reconstruction will be bounded by some constant fraction of the sparsity of the vector that cannot be made arbitrarily close to zero.

On the other hand, using a lossless condenser enables us to bring down the number of false positives in the reconstruction to an arbitrarily small fraction of d (which is, in light of Lemma 4.8, the best we can hope for), though it does not give as good a control on the fraction of tolerable errors as in the extractor case, though we still obtain resilience against the same order of errors.

Recall that the simple divide-and-conquer adaptive construction given in beginning the chapter consists of $O(\log(n/d))$ *non-adaptive stages*, where within each stage $O(d)$ non-adaptive measurements are made, but the choice of the measurements for each stage fully depends on all the previous outcomes. By the lower bounds on the size of disjunct matrices, we know that the number of non-adaptive rounds cannot be reduced to 1 without affecting the total number of measurements by a multiplicative factor of $\tilde{\Omega}(d)$. However, our non-adaptive upper bounds (Corollaries 4.16 and 4.17) show that the number of rounds can be reduced to 2, while preserving the total number of measurements at $O(d \log n)$. In particular, in a two-stage scheme, the first non-adaptive round would output an approximation of the d-sparse vector up to $O(d)$ false positive (even if the measurements are highly unreliable) and the second round simply examines the $O(d)$ possible positions using trivial singleton measurements to pinpoint the exact support of the vector.

Applying the Guruswami-Umans-Vadhan's Extractor

While Corollaries 4.16 and 4.17 give probabilistic constructions of noise-resilient measurement matrices, certain applications require a fully explicit matrix that is guaranteed to work. To that end, we need to instantiate Theorem 4.15 with an explicit condenser. First, we use the nearly-optimal explicit extractor of Guruswami, Umans and Vadhan (Theorem 2.24), that currently gives the best trade-off for the range of parameters needed for our application. Using this extractor, we obtain a similar trade-off as in Corollary 4.16, except for a higher number of measurements which would be bounded by $O(2^{O(\log^2 \log d)} d \log n) = O(d^{1+o(1)} \log n)$.

Corollary 4.18. *For every choice of constants $p \in [0, 1)$ and $\nu \in [0, \nu_0)$, $\nu_0 := (\sqrt{5 - 4p} - 1)^3/8$, and positive integers d and $n \geq d$, there is a fully explicit $m \times n$ measurement matrix, where*

$$m = O(2^{O(\log^2 \log d)} d \log n) = O(d^{1+o(1)} \log n),$$

that is $(pm, (\nu/d)m, O(d), 0)$-resilient for d-sparse vectors of length n and allows for a reconstruction algorithm with running time $O(mn)$. □

Applying "Zig-Zag" Lossless Condenser

An important explicit construction of lossless condensers that has an almost optimal output length is due to Capalbo et al. [23]. This construction borrows the notion of "zig-zag products" that is a combinatorial tool for construction of expander graphs as a major ingredient of the condenser. The following theorem quotes a setting of this construction that is most useful for our application:

Theorem 4.19. [23] *For every $k \leq n \in \mathbb{N}$, $\epsilon > 0$ there is an explicit $k \to_\epsilon k$ condenser*[5] *with seed length $d = O(\log^3(n/\epsilon))$ and output length $m = k + \log(1/\epsilon) + O(1)$.* □

Combining Theorem 4.15 with the above condenser, we obtain a similar result as in Corollary 4.17, except that the number of measurements would be $d2^{\log^3(\log n)} = d \cdot \mathsf{quasipoly}(\log n)$.

Corollary 4.20. *For positive integers $n \geq d$ and every constant $\delta > 0$ there is a fully explicit $m \times n$ measurement matrix, where*

$$m = d2^{\log^3(\log n)} = d \cdot \mathsf{quasipoly}(\log n),$$

that is $(\Omega(m), \Omega(1/d)m, \delta d, 0)$-resilient for d-sparse vectors of length n and allows for a reconstruction algorithm with running time $O(mn)$. □

4.2.2.3 Measurements Allowing Sublinear Time Reconstruction

The naive reconstruction algorithm given by Theorem 4.15 works efficiently in linear time in the size of the measurement matrix. However, for very sparse vectors (i.e., $d \ll n$), it might be of practical importance to have a reconstruction algorithm that runs in *sublinear* time in n, the length of the vector, and ideally, polynomial in the number of measurements, which is merely $\mathsf{poly}(\log n, d)$ if the number of measurements is optimal.

As shown in [149], if the code $\mathcal{C}$ in Theorem 4.14 is obtained from a strong extractor constructed from a *black-box pseudorandom generator (PRG)*, it is possible to compute the agreement list (which is guaranteed by the theorem to be small) more efficiently than a simple exhaustive search over all possible codewords. In particular, in this case they show that $\mathsf{LIST}_{\mathcal{C}}(S, \rho(S)+\epsilon)$ can be computed in time $\mathsf{poly}(2^t, 2^{\tilde{\ell}}, 2^k, 1/\epsilon)$ (where $t, \tilde{\ell}, k, \epsilon$ are respectively the seed length, output length, entropy requirement, and error of the extractor), which can be much smaller than $2^{\tilde{n}}$ ($\tilde{n}$ being the input length of the extractor).

Currently two constructions of extractors from black-box PRGs are known: Trevisan's extractor [152] (as well as its improvement in [123]) and Shaltiel-Umans' extractor [134]. However, the latter can only extract a sub-constant fraction of the min-entropy and is not suitable for our needs, albeit it requires a considerably shorter seed than Trevisan's extractor. Thus, here we only consider Raz's improvement of Trevisan's extractor given in Theorem 2.20. Using this extractor in Theorem 4.15, we obtain a measurement matrix for which the reconstruction is possible in polynomial time in the number of measurements; however, as the seed length required by this extractor is larger than Theorem 2.24, we will now require a higher number of measurements than before. Specifically, using Trevisan's extractor, we get the following.

[5] Though not explicitly mentioned in [23], these condensers can be considered to be strong.

Corollary 4.21. *For every choice of constants $p \in [0,1)$ and $\nu \in [0,\nu_0)$, $\nu_0 := (\sqrt{5-4p}-1)^3/8$, and positive integers d and $n \geq d$, there is a fully explicit $m \times n$ measurement matrix M that is $(pm,(\nu/d)m,O(d),0)$-resilient for d-sparse vectors of length n, where*

$$m = O(d2^{\log^3 \log n}) = d \cdot \mathsf{quasipoly}(\log n).$$

Furthermore, M allows for a reconstruction algorithm with running time $\mathsf{poly}(m)$, which would be sublinear in n for $d = O(n^c)$ and a suitably small constant $c > 0$. □

On the condenser side, we observe that the strong lossless (and lossy) condensers due to Guruswami et al. (given in Theorem 2.22) also allow efficient list-recovery. The code induced by this condenser is precisely a list-decodable code due to Parvaresh and Vardy [118]. Thus, the efficient list recovery algorithm of the condenser is merely the list-decoding algorithm for this code[6]. Combined with Theorem 4.15, we can show that codeword graphs of Parvaresh-Vardy codes correspond to good measurement matrices that allow sublinear time recovery, but with incomparable parameters to what we obtained from Trevisan's extractor (the proof is similar to Corollary 4.17):

Corollary 4.22. *For positive integers $n \geq d$ and any constants $\delta, \alpha > 0$ there is an $m \times n$ measurement matrix, where*

$$m = O(d^{3+\alpha+2/\alpha}(\log n)^{2+2/\alpha}),$$

that is $(\Omega(e),\Omega(e/d),\delta d,0)$-resilient for d-sparse vectors of length n, where

$$e := (\log n)^{1+1/\alpha} d^{2+1/\alpha}.$$

Moreover, the matrix allows for a reconstruction algorithm with running time $\mathsf{poly}(m)$. □

We remark that we could also use a lossless condenser due to Ta-Shma et al. [148] which is based on Trevisan's extractor and also allows efficient list recovery, but it achieves inferior parameters compared to Corollary 4.22.

4.2.2.4 Connection with List-Recoverability

Extractor codes that we used in Theorem 4.15 are instances of *soft-decision decodable* codes[7] that provide high list-decodability in "extremely noisy" scenarios. In fact it is not hard to see that good extractors or condensers are

[6] For similar reasons, any construction of measurement matrices based on codeword graphs of algebraic codes that are equipped efficient soft-decision decoding (including the original Reed-Solomon based construction of Kautz and Singleton [89]) allow sublinear time reconstruction.

[7] To be precise, here we are dealing with a special case of soft-decision decoding with binary weights.

required for our construction to carry through, as Theorem 4.14 can be shown to hold, up to some loss in parameters, in the reverse direction as well (as already shown by Ta-Shma and Zuckerman [149, Theorem 1] for the case of extractors).

However, for designing measurement matrices for the noiseless (or low-noise) case, it is possible to resort to the slightly weaker notion of *list recoverable codes*. Formally, a code $\mathcal{C}$ of block length $\tilde{n}$ over an alphabet Σ is called $(\alpha, d, \tilde{\ell})$-*list recoverable* if for every mixture S over $\Sigma^{\tilde{n}}$ consisting of sets of size at most d each, we have $|\mathsf{LIST}_{\mathcal{C}}(S, \alpha)| \leq \tilde{\ell}$. A simple argument similar to Theorem 4.15 shows that the adjacency matrix of the codeword graph of such a code with rate R gives a $(\log n)|\Sigma|/R \times n$ measurement matrix[8] for d-sparse vectors in the noiseless case with at most $\tilde{\ell} - d$ false positives in the reconstruction.

Ideally, a list-recoverable code with $\alpha = 1$, alphabet size $O(d)$, positive constant rate, and list size $\tilde{\ell} = O(d)$ would give an $O(d \log n) \times n$ matrix for d-sparse vectors, which is almost optimal (furthermore, the recovery would be possible in sublinear time if $\mathcal{C}$ is equipped with efficient list recovery). However, no explicit construction of such a code is so far known.

Two natural choices of codes with good list-recoverability properties are Reed-Solomon and Algebraic-Geometric codes, which in fact provide soft-decision decoding with short list size (cf. [74]). However, while the list size is polynomially bounded by $\tilde{n}$ and d, it can be much larger than $O(d)$ that we need for our application even if the rate is polynomially small in d.

On the other hand, it is shown in [77] that *folded Reed-Solomon Codes* are list-recoverable with constant rate, but again they suffer from large alphabet and list size[9].

We also point out a construction of (α, d, d) list-recoverable codes (allowing list recovery in time $O(\tilde{n}d)$) in [77] with rate polynomially small but alphabet size exponentially large in d, from which they obtain superimposed codes.

4.2.2.5 Connection with the Bit-Probe Model and Designs

An important problem in data structures is the static set membership problem in bit-probe model, which is the following: Given a set S of at most d elements from a universe of size n, store the set as a string of length m such that any query of the type "is x in S?" can be reliably answered by reading few bits of the encoding. The query algorithm might be probabilistic, and be allowed to err with a small one or two-sided error. Information theoretically, it is easy to

[8] For codes over large alphabets, the factor $|\Sigma|$ in the number of rows can be improved using *concatenation* with a suitable *inner* measurement matrix.

[9] As shown in [78], folded Reed-Solomon codes can be used to construct lossless condensers, which eliminates the list size problem. However, they give inferior parameters compared to Parvaresh-Vardy codes used in Corollary 4.22.

see that $m = \Omega(d \log(n/d))$ regardless of the bit-probe complexity and even if a small constant error is allowed.

Remarkably, it was shown in [19] that the lower bound on m can be (non-explicitly) achieved using only one bit-probe. Moreover, a part of their work shows that any one-probe scheme with negative one-sided error ϵ (where the scheme only errs in case $x \notin S$) gives a $\lfloor d/\epsilon \rfloor$-superimposed code (and hence, requires $m = \Omega(d^2 \log n)$ by [54]). It follows that from any such scheme one can obtain a measurement matrix for exact reconstruction of sparse vectors, which, by Lemma 4.8, cannot provide high resiliency against noise. The converse direction, i.e., using superimposed codes to design bit-probe schemes does not necessarily hold unless the error is allowed to be very close to 1. However, in [19] *combinatorial designs*[10] based on low-degree polynomials are used to construct one bit-probe schemes with $m = O(d^2 \log^2 n)$ and small one-sided error.

On the other hand, Kautz and Singleton [89] observed that the encoding of a combinatorial design as a binary matrix corresponds to a superimposed code (which is in fact slightly error-resilient). Moreover, they used Reed-Solomon codes to construct a design, which in particular gives a d-superimposed code. This is in fact the same design that is used in [19], and in our terminology, can be regarded as the adjacency matrix of the codeword graph of a Reed-Solomon code.

It is interesting to observe the intimate similarity between our framework given by Theorem 4.15 and classical constructions of superimposed codes. However, some key differences are worth mentioning. Indeed, both constructions are based on codeword graphs of error-correcting codes. However, classical superimposed codes owe their properties to the large distance of the underlying code. On the other hand, our construction uses extractor and condenser codes and does not give a superimposed code simply because of the substantially low number of measurements. However, as shown in Theorem 4.15, they are good enough for a slight relaxation of the notion of superimposed codes because of their soft-decision list decodability properties, which additionally enables us to attain high noise resilience and a considerably smaller number of measurements.

Interestingly, Buhrman et al. [19] use randomly chosen bipartite graphs to construct storage schemes with two-sided error requiring nearly optimal space $O(d \log n)$, and Ta-Shma [147] later shows that expander graphs from lossless condensers would be sufficient for this purpose. However, unlike schemes with negative one-sided error, these schemes use encoders that cannot be implemented by the "or" function and thus do not translate to group testing schemes.

[10] A design is a collection of subsets of a universe, each of the same size, such that the pairwise intersection of any two subset is upper bounded by a prespecified parameter.

4.3 The Threshold Model

A natural generalization of classical group testing, introduced by Damaschke [42], considers the case where the measurement outcomes are determined by a *threshold predicate* instead of logical "or".

In particular, the threshold model is characterized by two integer parameters ℓ, u such that $0 < \ell \leq u$ (that are considered to be fixed constants), and each measurement outputs positive if the number of positives within the corresponding pool is at least u. On the other hand, if the number of positives is less than ℓ, the test returns negative, and otherwise the outcome can be arbitrary. In this view, classical group testing corresponds to the special case where $\ell = u = 1$. In addition to being of theoretical interest, the threshold model is interesting for applications, in particular in biology, where the measurements have reduced or unpredictable sensitivity or may depend on various factors that must be simultaneously present in the sample.

The difference $g := u - \ell$ between the thresholds is known as the *gap* parameter. As shown by Damaschke [42], in threshold group testing identification of the set of positives is only possible when the number of positives is at least u. Moreover, regardless of the number of measurements, in general the set of positives can only be identified within up to g false positives and g false negatives (thus, unique identification can be guaranteed only when $\ell = u$).

Additionally, Damaschke constructed a scheme for identification of the positives in the threshold model. For the gap-free case where $g = 0$, the number of measurements in this scheme is $O((d + u^2) \log n)$, which is nearly optimal (within constant factors). However, when $g > 0$, the number of measurements becomes $O(dn^b + d^u)$, for an arbitrary constant $b > 0$, if up to $g + (u - 1)/b$ misclassifications are allowed. Moreover, Chang et al. [24] have proposed a different scheme for the gap-free case that achieves $O(d \log n)$ measurements.

A drawback of the scheme presented by Damaschke (as well as the one by Chang et al.) is that the measurements are adaptive. As mentioned before, for numerous applications (in particular, molecular biology), adaptive measurements are infeasible and must be avoided.

In this section, we consider the non-adaptive threshold testing problem in a possibly noisy setting, and develop measurement matrices that can be used in the threshold model. Similar to the classical model of group testing, non-adaptive measurements in the threshold model can be represented as a Boolean matrix, where the ith row is the characteristic vector of the set of items that participate in the ith measurement.

4.3.1 Strongly Disjunct Matrices

Non-adaptive threshold testing has been considered by Chen and Fu [27]. They observe that, a generalization of the standard notion of disjunct matrices

(the latter being extensively used in the literature of classical group testing) is suitable for the threshold model. In this section, we refer to this generalized notion as *strongly disjunct* matrices and to the standard notion as *classical* disjunct matrices. Strongly disjunct matrices can be defined as follows.

Definition 4.23. A Boolean matrix (with at least $d+u$ columns) is said to be strongly $(d, e; u)$-disjunct if for every choice of $d+u$ distinct columns

$$C_1, \ldots, C_u, C'_1, \ldots, C'_d,$$

all distinct, we have

$$|\cap_{i=1}^{u} \mathsf{supp}(C_i) \setminus \cup_{i=1}^{d} \mathsf{supp}(C'_i)| > e.$$

Observe that, $(d, e; u)$-disjunct matrices are, in particular, $(d', e'; u')$-disjunct for any $d' \leq d$, $e' \leq e$, and $u' \leq u$. Moreover, *classical* (d, e)-disjunct matrices correspond to the special case $u = 1$.

An important motivation for the study of this notion is the following *hidden hypergraph learning problem* (cf. [51, Chapter 6] and [50, Chapter 12]), itself being motivated by the so-called *complex model* in computational biology [26]. A $(\leq u)$-hypergraph is a tuple (V, E) where V and E are known as the set of vertices and hyper-edges, respectively. Each hyperedge $e \in E$ is a non-empty subset of V of size at most u. The classical notion of undirected graphs (with self-loops) corresponds to (≤ 2)-hypergraphs.

Now, suppose that G is a $(\leq u)$-hypergraph on a vertex set V of size n, and denote by $\mathcal{V}(G)$ the set of vertices induced by the hyper-edge set of G; i.e., $v \in \mathcal{V}(G)$ if and only if G has a hyper-edge incident to v. Then assuming that $|\mathcal{V}(G)| \leq d$ for a *sparsity parameter* d, the aim in the hypergraph-learning problem is to identify G using as few (non-adaptive) queries of the following type as possible: Each query specifies a set $Q \subseteq V$, and its corresponding answer is a Boolean value which is 1 if and only if G has a hyperedge contained in Q.

It is known that [26, 66], in the hypergraph learning problem, any suitable grouping strategy defines a strongly disjunct matrix (whose rows are characteristic vectors of individual queries Q), and conversely, any strongly disjunct matrix can be used as the incidence matrix of the set of queries. Below we recollect a simple proof of this fact.

Lemma 4.24. *Let M be a strongly $(d, e; u)$-disjunct matrix with columns indexed by the elements of a vertex set V, and G and G' be any two distinct $(\leq u)$-hypergraphs on V such that $\mathcal{V}(G) \leq d$ and $\mathcal{V}(G') \leq d$. Then the vector of the outcomes corresponding to the queries defined by M on G and G' differ in more than e positions. Conversely, if M is such that the query outcomes differ in more than e positions for every choice of the hypergraphs G and G' as above, then it must be strongly $(d-u, e; u)$-disjunct.*

Proof. Suppose that M is an $m \times |V|$ strongly $(d, e; u)$-disjunct matrix, and consider distinct $(\leq u)$-hypergraphs $G = (V, E)$ and $G' = (V, E')$ with $\mathcal{V}(G) \leq d$ and $\mathcal{V}(G') \leq d$. Denote by $y, y' \in \{0,1\}^m$ the vector of query outcomes for the two graphs G and G', respectively. Without loss of generality, let $S \in E$ be chosen such that no hyper-edge of G' is contained in it. Let $V' := \mathcal{V}(G') \setminus S$, and denote by $C_1, \ldots, C_{|S|}$ (resp., $C'_1, \ldots, C_{|V'|}$) the columns of M corresponding to the vertices in S (resp., V'). By Definition 4.23, there is a set $T \subseteq [m]$ of more than e indices such that for every $i \in [|S|]$ (resp., $i \in [|V'|]$) and every $t \in T$, $C_i(t) = 1$ (resp., $C'_i(t) = 0$). This means that, for each such t, the answer to the tth query must be 1 for G (as the query includes the vertex set of S) but 0 for G' (considering the assumption that no edge of G' is contained in S).

For the converse, let $S, Z \subseteq [V]$ be disjoint sets of vertices such that $|S| = u$ and $|Z| = d - u$, and denote by $\{C_1, \ldots, C_u\}$ and $\{C'_1, \ldots, C'_{d-u}\}$ the set of columns of M picked by S and T, respectively. Take any $v \in S$, let the u-hypergraph $G = (V, E)$ be a u-clique on $Z \cup S \setminus \{v\}$, and $G' = (V, E')$ be such that $E' := E \cup \{S\}$. Denote by $y, y' \in \{0,1\}^m$ the vector of query outcomes for the two graphs G and G', respectively. Since G' is a subgraph of G, it must be that $\mathsf{supp}(y') \subseteq \mathsf{supp}(y)$.

Let $T := \mathsf{supp}(y) \setminus \mathsf{supp}(y)$. By the distinguishing property of M, the set T must have more than e elements. Take any $t \in T$. We know that the tth query defined by M returns positive for G but negative for G'. Thus this query must contain the vertex set of S, but not any of the elements in Z (since otherwise, it would include some $z \in Z$ and subsequently, $\{z\} \cup S \setminus \{v\}$, which is a hyperedge of G'). It follows that for each $i \in [u]$ (resp., $i \in [d-u]$), we must have $C_i(t) = 1$ (resp., $C'_i(t) = 0$) and the disjunctness property as required by Definition 4.23 holds. □

The parameter e determines "noise tolerance" of the measurement scheme. Namely, a strongly $(d, e; u)$-disjunct matrix can uniquely distinguish between d-sparse hypergraphs even in presence of up to $\lfloor e/2 \rfloor$ erroneous query outcomes.

The key observation made by Chen and Fu [27] is that threshold group testing corresponds to the special case of the hypergraph learning problem where the hidden graph G is known to be a u-clique[11]. In this case, the unknown Boolean vector in the corresponding threshold testing problem would be the characteristic vector of $\mathcal{V}(G)$. It follows that strongly disjunct matrices are suitable choices for the measurement matrices in threshold group testing.

More precisely, the result by Chen and Fu states that, for threshold parameters ℓ and u, a strongly $(d-\ell-1, 2e; u)$-disjunct matrix suffices to distinguish

[11] A u-clique on the vertex set V is a $(\leq u)$-hypergraph (V, E) such that, for some $V' \subseteq V$, E is the set of all subsets of V' of size u.

between d-sparse vectors in the threshold model[12], even if up to e erroneous measurements are allowed.

Much of the known results for classical disjunct matrices can be extended to strongly disjunct matrices by following similar ideas. In particular, the probabilistic result of Theorem 4.4 can be generalized to show that strongly $(d, e; u)$-disjunct matrices exist with

$$m = O(d^{u+1}(\log(n/d))/(1-p)^2)$$

rows and error tolerance

$$e = \Omega(pd\log(n/d)/(1-p)^2),$$

for any noise parameter $p \in [0, 1)$. On the negative side, however, several concrete lower bounds are known for the number of rows of such matrices [53, 144, 145]. In asymptotic terms, these results show that one must have

$$m = \Omega(d^{u+1}\log_d n + ed^u),$$

and thus, the probabilistic upper bound is essentially optimal.

4.3.2 Strongly Disjunct Matrices from Codes

For the underlying strongly disjunct matrix, Chen and Fu [27] use a greedy construction [28] that achieves, for any $e \geq 0$, $O((e+1)d^{u+1}\log(n/d))$ rows, but may take exponential time in the size of the resulting matrix.

Nevertheless, as observed by several researchers [26, 53, 66, 91], a classical explicit construction of combinatorial designs due to Kautz and Singleton [89] can be extended to construct strongly disjunct matrices. This concatenation-based construction transforms any error-correcting code having large distance into a disjunct matrix.

While the original construction of Kautz and Singleton uses Reed-Solomon codes and achieves nice bounds, it is possible to use other families of codes. In particular, as was shown by Porat and Rothschild [120], codes on the Gilbert-Varshamov bound (see Appendix A) would result in nearly optimal disjunct matrices. Moreover, for a suitable range of parameters, they give a *deterministic* construction of such codes that runs in polynomial time in the size of the resulting disjunct matrix (albeit exponential in code's dimension[13]).

In this section, we will elaborate on details of this (known) class of constructions, and in addition to Reed-Solomon codes and codes on the Gilbert-Varshamov bound (that, as mentioned above, were used by Kautz, Singleton,

[12] Considering unavoidable assumptions that up to $g := u - \ell$ false positives and g false negatives are allowed in the reconstruction, and that the vector being measured has weight at least u.

[13] In this regard, this construction of disjunct matrices can be considered *weakly explicit* in that, contrary to fully explicit constructions, it is not clear if each individual entry of the matrix can be computed in time $\mathsf{poly}(d, \log n)$.

- *Given:* An $(\tilde{n}, k, \tilde{d})_q$ error-correcting code $\mathcal{C} \subseteq [q]^{\tilde{n}}$, and integer parameter $u > 0$.
- *Output:* An $m \times n$ Boolean matrix M, where $n = q^k$, and $m = \tilde{n}q^u$.
- *Construction:* First, consider the mapping $\varphi\colon [q] \to \{0,1\}^{q^u}$ from q-ary symbols to column vectors of length q^u defined as follows. Index the coordinates of the output vector by the u-tuples from the set $[q]^u$. Then $\varphi(x)$ has a 1 at position $(a_1, \ldots, a_u)$ if and only if there is an $i \in [u]$ such that $a_i = x$. Arrange all codewords of $\mathcal{C}$ as columns of an $\tilde{n} \times q^k$ matrix M' with entries from $[q]$. Then replace each entry x of M' with $\varphi(x)$ to obtain the output $m \times n$ matrix M.

Construction 4.1: Extension of Kautz-Singleton's method [89].

Porat and Rothschild), will consider a family of algebraic-geometric codes and Hermitian codes which give nice bounds as well. Construction 4.1 describes the general idea, which in analyzed in the following lemma.

Lemma 4.25. *Construction 4.1 outputs a strongly $(d, e; u)$-disjunct matrix for every $d < (\tilde{n} - e)/((\tilde{n} - \tilde{d})u)$.*

Proof. Let $C := \{c_1, \ldots, c_u\} \subseteq [n]$ and $C' := \{c'_1, \ldots, c'_d\} \subseteq [n]$ be disjoint subsets of column indices. We wish to show that, for more than e rows of M, the entries at positions picked by C are all-ones while those picked by C' are all-zeros. For each $j \in [n]$, denote the jth column of M' by $M'(j)$, and let $M'(C) := \{M'(c_j)\colon j \in [u]\}$, and $M'(C') := \{M'(c'_j)\colon j \in [d]\}$.

From the minimum distance of $\mathcal{C}$, we know that every two distinct columns of M' agree in at most $\tilde{n} - \tilde{d}$ positions. By a union bound, for each $i \in [d]$, the number of positions where $M'(c'_i)$ agrees with one or more of the codewords in $M'(C)$ is at most $u(\tilde{n} - \tilde{d})$, and the number of positions where some vector in $M'(C')$ agrees with one or more of those in $M'(C)$ is at most $du(\tilde{n} - \tilde{d})$. By assumption, we have $\tilde{n} - du(\tilde{n} - \tilde{d}) > e$, and thus, for a set $E \subseteq [\tilde{n}]$ of size greater than e, at positions picked by E none of the codewords in $M'(C')$ agree with any of the codewords in $M'(C)$.

Now let $w \in [q]^n$ be any of the rows of M' picked by E, and consider the $q^u \times n$ Boolean matrix W formed by applying the mapping $\varphi(\cdot)$ on each entry of w. We know that $\{w(c_j)\colon j \in [u]\} \cap \{w(c'_j)\colon j \in [d]\} = \emptyset$. Thus we observe that the particular row of W indexed by $(w(c_1), \ldots, w(c_u))$ (and in fact, any of its permutations) must have all-ones at positions picked by C and all-zeros at those picked by C'. As any such row is a distinct row of M, it follows that M is strongly $(d, e; u)$-disjunct. □

Now we mention a few specific instantiations of the above construction. We will first consider the family of Reed-Solomon codes, that are also used in the original work of Kautz and Singleton [89], and then move on to the family of algebraic geometric (AG) codes on the Tsfasman-Vlăduţ-Zink (TVZ) bound, and Hermitian codes, and finally, codes on the Gilbert-Varshamov (GV) bound. A quick review of the necessary background on coding-theoretic terms is given in Appendix A.

Reed-Solomon Codes

Let $p \in [0,1)$ be an arbitrary "noise" parameter. If we take $\mathcal{C}$ to be an $[\tilde{n}, k, \tilde{d}]_{\tilde{n}}$ Reed-Solomon code over an alphabet of size $\tilde{n}$ (more precisely, the smallest prime power that is no less than $\tilde{n}$), where $\tilde{d} = \tilde{n} - k + 1$, we get a strongly disjunct $(d, e; u)$-matrix with

$$m = O(du \log n/(1-p))^{u+1}$$

rows and

$$e = p\tilde{n} = \Omega(pdu(\log n)/(1-p)).$$

AG Codes on the TVZ Bound

Another interesting family for the code $\mathcal{C}$ is the family of algebraic geometric codes that attain the Tsfasman-Vlăduţ-Zink bound (cf. [67, 154]). This family is defined over any alphabet size $q \geq 49$ that is a square prime power, and achieves a minimum distance $\tilde{d} \geq \tilde{n} - k - \tilde{n}/(\sqrt{q} - 1)$. Let $e := pn$, for a noise parameter $p \in [0,1)$. By Lemma 4.25, the underlying code $\mathcal{C}$ needs to have minimum distance at least $\tilde{n}(1 - (1-p)/(du))$. Thus in order to be able to use the above-mentioned family of AG codes, we need to have $q \gg (du/(1-p))^2 =: q_0$. Let us take an appropriate $q \in [2q_0, 8q_0]$, and following Lemma 4.25, $\tilde{n} - \tilde{d} = \lceil \tilde{n}(1-p)/(du) \rceil$. Thus the dimension of $\mathcal{C}$ becomes at least

$$k \geq \tilde{n} - \tilde{d} - \frac{\tilde{n}}{\sqrt{q} - 1} = \Omega\left(\frac{\tilde{n}(1-p)}{du}\right) = \Omega(\tilde{n}/\sqrt{q_0}),$$

and subsequently[14] we get that $\log n = k \log q \geq k = \Omega(\tilde{n}/\sqrt{q_0})$. Now, noting that $m = q^u \tilde{n}$, we conclude that

$$m = q^u \tilde{n} = O(q_0^{u+1/2} \log n) = O\left(\frac{du}{1-p}\right)^{2u+1} \log n,$$

and $e = \Omega(pdu(\log n)/(1-p))$.

[14] Note that, given the parameters p, d, n, the choice of q depends on p, d, as explained above, and then one can choose the code length $\tilde{n}$ to be the smallest integer for which we have $q^k \geq n$. But for the sake of clarity we have assumed that $q^k = n$.

We see that the dependence of the number of measurements on the sparsity parameter d is worse for AG codes than Reed-Solomon codes by a factor d^u, but the construction from AG codes benefits from a linear dependence on $\log n$, compared to $\log^{u+1} n$ for Reed-Solomon codes. Thus, AG codes become more favorable only when the sparsity is substantially low; namely, when $d \ll \log n$.

Hermitian Codes

A particularly nice family of AG codes arises from the Hermitian function field[15]. Let q' be a prime power and $q := q'^2$. Then the Hermitian function field over $\mathbb{F}_q$ is a finite extension of the rational function field $\mathbb{F}_q(x)$, denoted by $\mathbb{F}_q(x, y)$, where we have $y^{q'} + y = x^{q'+1}$. The structure of this function field is relatively well understood and the family of Goppa codes defined over the rational points of the Hermitian function field is known as Hermitian codes. This family is recently used by Ben-Aroya and Ta-Shma [10] for construction of small-bias sets. Below we quote some parameters of Hermitian codes from their work.

The number of rational points of the Hermitian function field is equal to $q'^3 + 1$, which includes a common pole Q_∞ of x and y. The genus of the function field is $g = q'(q' - 1)/2$. For some integer parameter r, we take $G := rQ_\infty$ as the divisor defining the Riemann-Roch space $\mathcal{L}(G)$ of the code $\mathcal{C}$, and the set of rational points except Q_∞ as the evaluation points of the code. Thus the length of $\mathcal{C}$ becomes $\tilde{n} = q'^3$. Moreover, the minimum distance of the code is $\tilde{d} = n - \deg(G) = n - r$. When $r \geq 2g - 1$, the dimension of the code is given by the Riemann-Roch theorem, which is equal to $r - g + 1$. For the low-degree regime where $r < 2g - 1$, the dimension k of the code is the size of the Wirestrauss semigroup of G, which turns out to be the set $W = \{(i, j) \in \mathbb{N}^2 \colon j \leq q' - 1 \wedge iq' + j(q' + 1) \leq r\}$.

Now, given parameters d, p of the disjunct matrix, define $\rho := (1-p)/((d+1)u)$, take the alphabet size q as a square prime power, and set $r := \rho q^{3/2}$. First we consider the case where $r < 2g - 1 = 2q - 2\sqrt{q} - 1$. In this case, the dimension of the Hermitian code becomes $k = |W| = \Omega(r^2/q) = \Omega(\rho^2 q^2)$. The distance $\tilde{d}$ of the code satisfies $\tilde{d} = \tilde{n} - r \geq \tilde{n}(1 - \rho)$ and thus, for $e := p\tilde{n}$, conditions of Lemma 4.25 are satisfied. The number of the rows of the resulting measurement matrix becomes $m = q^{u+3/2}$, and we have $n = q^k$. Therefore,

$$\log n = k \log q \geq k = \Omega(\rho^2 q^2)$$
$$\Rightarrow q = O(\sqrt{\log n}/\rho) \Rightarrow m = O\left(\left(\frac{d\sqrt{\log n}}{1-p} \right)^{u+3/2} \right),$$

and in order to ensure that $r < 2g - 1$, we need to have $du/(1 - p) \gg \sqrt{\log n}$. On the other hand, when $du/(1 - p) \ll \sqrt{\log n}$, we are in the high-degree

[15] See [142] for an extensive treatment of the notions in algebraic geometry.

regime, in which case the dimension of the code becomes $k = r - g + 1 = \Omega(r) = \Omega(\rho q^{3/2})$, and we will thus have

$$q = O((\log n/\rho)^{2/3}) \Rightarrow m = O\left((\frac{d \log n}{1-p})^{1+2u/3}\right)$$

Altogether, we conclude that Construction 4.1 with Hermitian codes results in a strongly $(d, e; u)$-disjunct matrix with

$$m = O\left((\frac{d\sqrt{\log n}}{1-p} + (\frac{d \log n}{1-p})^{2/3})^{u+3/2}\right)$$

rows, where $e = p \cdot \Omega\left(d(\log n)/(1-p) + (d\sqrt{\log n}/(1-p))^{3/2}\right)$. Compared to the Reed-Solomon codes, the number of measurements has a slightly worse dependence on d, but a much better dependence on n. Compared to AG codes on the TVZ bound, the dependence on d is better while the dependence on n is inferior.

Codes on the GV Bound

A q-ary $(\tilde{n}, k, \tilde{d})$-code (of sufficiently large length) is said to be on the Gilbert-Varshamov bound if it satisfies $k \geq \tilde{n}(1 - h_q(\tilde{d}/\tilde{n}))$, where $h_q(\cdot)$ is the q-ary entropy function defined as

$$h_q(x) := x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x).$$

It is well known that a random linear code achieves the bound with overwhelming probability (cf. [103]). Now we apply Lemma 4.25 on a code on the GV bound, and calculate the resulting parameters. Let $\rho := (1-p)/(4du)$, choose any alphabet size $q \in [1/\rho, 2/\rho]$, and let $\mathcal{C}$ be any q-ary code of length $\tilde{n}$ on the GV bound, with minimum distance $\tilde{d} \geq \tilde{n}(1 - 2/q)$. By the Taylor expansion of the function $h_q(x)$ around $x = 1 - 1/q$, we see that the dimension of $\mathcal{C}$ asymptotically behaves as $k = \Theta(\tilde{n}/(q \log q))$. Thus the number of columns of the resulting measurement matrix becomes $n = q^k = 2^{\Omega(\tilde{n}/q)}$, and therefore, the number m of its rows becomes

$$m = q^u \tilde{n} = O(q^{u+1} \log n) = O((d/(1-p))^{u+1} \log n),$$

and the matrix would be strongly $(d, e; u)$-disjunct for

$$e = p\tilde{n} = \Omega(pd(\log n)/(1-p)).$$

We remark that for the range of parameters that we are interested in, Porat and Rothschild [120] have recently come up with a deterministic construction of linear codes on the GV bound that runs in time $\mathsf{poly}(q^k)$ (and thus, polynomial in the size of the resulting measurement matrix). Their construction

Table 4.2: Bounds obtained by strongly $(d, e; u)$-disjunct matrices. The noise parameter $p \in [0, 1)$ is arbitrary. The first four rows correspond to the explicit coding-theoretic construction described in Section 4.3.2, with the underlying code indicated as a remark.

Number of rows	Noise tolerance	Remark
$O((\frac{d}{1-p})^{u+1} \log n)$	$\Omega(pd\frac{\log n}{1-p})$	Using codes on the GV bound.
$O((\frac{d \log n}{1-p})^{u+1})$	$\Omega(pd\frac{\log n}{1-p})$	Using Reed-Solomon codes.
$O((\frac{d}{1-p})^{2u+1} \log n)$	$\Omega(pd\frac{\log n}{1-p})$	Using Algebraic Geometric codes.
$O((\frac{d\sqrt{\log n}}{1-p})^{u+3/2})$	$\Omega(p(\frac{d\sqrt{\log n}}{1-p})^{3/2})$	Using Hermitian codes ($d \gg \sqrt{\log n}$).
$O(d^{u+1}\frac{\log(n/d)}{(1-p)^2})$	$\Omega(pd\frac{\log(n/d)}{(1-p)^2})$	Probabilistic construction.
$\Omega(d^{u+1} \log_d n + ed^u)$	e	Lower bound (Section 4.3.1).

is based on a derandomization of the probabilistic argument for random linear codes using the method of conditional expectations, and as such, can be considered *weakly explicit* (in the sense that, the entire measurement matrix can be computed in polynomial time in its length; but for a fully explicit construction one must be ideally able to deterministically compute any single entry of the measurement matrix in time $\mathsf{poly}(d, \log n)$, which is not the case for this construction).

We see that, for a fixed p, Construction 4.1 when using codes on the GV bound achieves almost optimal parameters. Moreover, the explicit construction based on the Reed-Solomon codes possesses the "right" dependence on the sparsity d, AG codes on the TVZ bound have a matching dependence on the vector length n with random measurement matrices, and finally, the trade-off offered by the construction based on Hermitian codes lies in between the one for Reed-Solomon codes and AG codes. These parameters are summarized in Table 4.2. Note that the special case $u = 1$ would give classical (d, e)-disjunct matrices as in Definition 4.1.

4.3.3 Disjunct Matrices for Threshold Testing

Even though, as discussed above, the general notion of strongly $(d, e; u)$-disjunct matrices is sufficient for threshold group testing with upper threshold u, in this section we show that a weaker notion of disjunct matrices (which turns out to be *strictly* weaker when the lower threshold ℓ is greater than 1), would also suffice. We proceed by showing how such measurement matrices can be constructed.

Before introducing our variation of disjunct matrices, let us fix some notation that will be useful for the threshold model. Consider the threshold model

with thresholds ℓ and u, and an $m \times n$ measurement matrix M that defines the set of measurements. For a vector $x \in \{0,1\}^n$, denote by $M[x]_{\ell,u}$ the set of vectors in $\{0,1\}^m$ that correctly encode the measurement outcomes corresponding to the vector x. In particular, for any $y \in M[x]_{\ell,u}$ we have $y(i) = 1$ if $|\mathsf{supp}(M_j) \cap \mathsf{supp}(x)| \geq u$, and $y(i) = 0$ if $|\mathsf{supp}(M_j) \cap \mathsf{supp}(x)| < \ell$, where M_j indicates the jth row of M. In the gap-free case, the set $M[x]_{\ell,u}$ may only have a single element that we denote by $M[x]_u$. Note that the gap-free case with $u = 1$ reduces to ordinary group testing, and thus we have $M[x]_1 = M[x]$.

To make the main ideas more transparent, until Section 4.3.3.3 we will focus on the gap-free case where $\ell = u$. The extension to nonzero gaps is straightforward and will be discussed in Section 4.3.3.3. Moreover, often we will implicitly assume that the Hamming weight of the Boolean vector that is to be identified is at least u (since otherwise, any $(u-1)$-sparse vector would be confused with the all-zeros vector). Moreover, we will take the thresholds ℓ, u as fixed constants while the parameters d and n are allowed to grow.

4.3.3.1 The Definition and Properties

Our variation of disjunct matrices along with an "auxiliary" notion of *regular* matrices is defined in the following.

Definition 4.26. A Boolean matrix M with n columns is called $(d, e; u)$-regular if for every subset of columns $S \subseteq [n]$ (called the *critical set*) and every $Z \subseteq [n]$ (called the *zero set*) such that $u \leq |S| \leq d$, $|Z| \leq |S|$, $S \cap Z = \emptyset$, there are more than e rows of M at which $M|_S$ has weight exactly u and (at the same rows) $M|_Z$ has weight zero. Any such row is said to *u-satisfy* S and Z.

If, in addition, for every *distinguished column* $i \in S$, more than e rows of M both u-satisfy S and Z and have a 1 at the ith column, the matrix is called $(d, e; u)$-disjunct (and the corresponding "good" rows are said to u-satisfy i, S, and Z).

It is easy to verify that (assuming $2d \leq n$) the classical notion of $(2d-1, e)$-disjunct matrices is equivalent to strongly $(2d-1, e; 1)$-disjunct and $(d, e; 1)$-disjunct. Moreover, any $(d, e; u)$-disjunct matrix is $(d, e; u)$-regular, $(d-1, e; u-1)$-regular, and (d, e)-disjunct (but the reverse implications do not in general hold). Therefore, the lower bound

$$m = \Omega(d^2 \log_d n + ed)$$

that applies for (d, e)-disjunct matrices holds for $(d, e; u)$-disjunct matrices as well.

Below we show that our notion of disjunct matrices is necessary and sufficient for the purpose of threshold group testing:

Lemma 4.27. *Let M be an $m \times n$ Boolean matrix that is $(d, e; u)$-disjunct. Then for every distinct d-sparse vectors $x, x' \in \{0,1\}^n$ such that*[16] $\mathsf{supp}(x) \not\subseteq \mathsf{supp}(x')$, $\mathsf{wgt}(x) \geq |\mathsf{supp}(x') \setminus \mathsf{supp}(x)|$ *and* $\mathsf{wgt}(x) \geq u$, *we have*

$$|\mathsf{supp}(M[x]_u) \setminus \mathsf{supp}(M[x']_u)| > e. \tag{4.5}$$

Conversely, assuming $d \geq 2u$, if M satisfies (4.5) *for every choice of x and x' as above, it must be $(\lfloor d/2 \rfloor, e; u)$-disjunct.*

Proof. First, suppose that M is $(d, e; u)$-disjunct, and let $y := M[x]_u$ and $y' := M[x']_u$. Take any $i \in \mathsf{supp}(x) \setminus \mathsf{supp}(x')$, and let $S := \mathsf{supp}(x)$ and $Z := \mathsf{supp}(x') \setminus \mathsf{supp}(x)$. Note that $|S| \leq d$ and by assumption, we have $|Z| \leq |S|$. Now, Definition 4.26 implies that there is a set E of more than e rows of M that u-satisfy i as the distinguished column, S as the critical set and Z as the zero set. Thus for every $j \in E$, the jth row of M restricted to the columns chosen by $\mathsf{supp}(x)$ must have weight exactly u, while its weight on $\mathsf{supp}(x')$ is less than u. Therefore, $y(j) = 1$ and $y'(j) = 0$ for more than e choices of j.

For the converse, consider any choice of a distinguished column $i \in [n]$, a critical set $S \subseteq [n]$ containing i (such that $|S| \geq u$), and a zero set $Z \subseteq [n]$ where $|Z| \leq |S|$. Define d-sparse Boolean vectors $x, x' \in \{0,1\}^n$ so that $\mathsf{supp}(x) := S$ and $\mathsf{supp}(x') := S \cup Z \setminus \{i\}$. Let $y := M[x]_u$ and $y' := M[x']_u$ and $E := \mathsf{supp}(y) \setminus \mathsf{supp}(y')$. By assumption we know that $|E| > e$. Take any $j \in E$. Since $y(j) = 1$ and $y'(j) = 0$, we get that the jth row of M restricted to the columns picked by $S \cup Z \setminus \{i\}$ must have weight at most $u - 1$, whereas it must have weight at least u when restricted to S. As the sets $\{i\}, S \setminus \{i\}$, and Z are disjoint, this can hold only if $M[j, i] = 1$, and moreover, the jth row of M restricted to the columns picked by S (resp., Z) has weight exactly u (resp., zero). Hence, this row (as well as all the rows of M picked by E) must u-satisfy i, S, and Z, confirming that M is $(\lfloor d/2 \rfloor, e; u)$-disjunct. □

We will use regular matrices as intermediate building blocks in our constructions of disjunct matrices to follow. The connection with disjunct matrices is made apparent through a direct product of matrices defined in Construction 4.2. Intuitively, using this product, regular matrices can be used to transform any measurement matrix suitable for the standard group testing model to one with comparable properties in the threshold model. The following lemma formalizes this idea.

Lemma 4.28. *Let M_1 and M_2 be Boolean matrices with n columns, such that M_1 is $(d-1, e_1; u-1)$-regular. Let $M := M_1 \odot M_2$, and suppose that for d-sparse Boolean vectors $x, x' \in \{0,1\}^n$ such that $\mathsf{wgt}(x) \geq \mathsf{wgt}(x')$, we have*

$$|\mathsf{supp}(M_2[x]_1) \setminus \mathsf{supp}(M_2[x']_1)| \geq e_2.$$

[16] Note that at least one of the two possible orderings of any two distinct d-sparse vectors, at least one having weight u or more, satisfies this condition.

Then, $|\mathsf{supp}(M[x]_u) \setminus \mathsf{supp}(M[x']_u)| \geq (e_1+1)e_2$.

Proof. First we consider the case where $u > 1$. Let $y := M_2[x]_1 \in \{0,1\}^{m_2}$, $y' := M_2[x']_1 \in \{0,1\}^{m_2}$, where m_2 is the number of rows of M_2, and let $E := \mathsf{supp}(y) \setminus \mathsf{supp}(y')$. By assumption, $|E| \geq e_2$. Fix any $i \in E$ so that $y(i) = 1$ and $y'(i) = 0$. Therefore, the ith row of M_2 must have all zeros at positions corresponding to $\mathsf{supp}(x')$ and there is a $j \in \mathsf{supp}(x) \setminus \mathsf{supp}(x')$ such that $M_2[i,j] = 1$. Define $S := \mathsf{supp}(x) \setminus \{j\}$, $Z := \mathsf{supp}(x') \setminus \mathsf{supp}(x)$, $z := M[x]_u$ and $z' := M[x']_u$.

As $\mathsf{wgt}(x) \geq \mathsf{wgt}(x')$, we know that $|Z| \leq |S| + 1$. The extreme case $|Z| = |S| + 1$ only happens when x and x' have disjoint supports, in which case one can remove an arbitrary element of Z to ensure that $|Z| \leq |S|$ and the following argument (considering the assumption $u > 1$) still goes through. By the definition of regularity, there is a set E_1 consisting of at least $e_1 + 1$ rows of M_1 that $(u-1)$-satisfy the critical set S and the zero set Z. Pick any $k \in E_1$, and observe that z must have a 1 at position (k,i). This is because the row of M indexed by (k,i) has a 1 at the jth position (since the ith row of M_2 does), and at least $u-1$ more 1's at positions corresponding to $\mathsf{supp}(x) \setminus \{j\}$ (due to regularity of M_1). On the other hand, note that the kth row of M_1 has at most $u-1$ ones at positions corresponding to $\mathsf{supp}(x')$ (because $\mathsf{supp}(x') \subseteq S \cup Z$), and the ith row of M_2 has all zeros at those positions (because $y'(i) = 0$). This means that the row of M indexed by (k,i) (which is the bit-wise or of the kth row of M_1 and the ith row of M_2) must have less than u ones at positions corresponding to $\mathsf{supp}(x')$, and thus, z' must be 0 at position (k,i). Therefore, z and z' differ at position (k,i).

Since there are at least e_2 choices for i, and for each choice of i, at least $e_1 + 1$ choices for k, we conclude that in at least $(e_1+1)e_2$ positions, z has a one while z' has a zero.

The argument for $u = 1$ is similar, in which case it suffices to take $S := \mathsf{supp}(x)$ and $Z := \mathsf{supp}(x') \setminus \mathsf{supp}(x)$. □

- *Given:* Boolean matrices M_1 and M_2 that are $m_1 \times n$ and $m_2 \times n$, respectively.
- *Output:* An $m \times n$ Boolean matrix $M_1 \odot M_2$, where $m := m_1 m_2$.
- *Construction:* Let the rows of $M := M_1 \odot M_2$ be indexed by the set $[m_1] \times [m_2]$. Then the row corresponding to (i,j) is defined as the bit-wise or of the ith row of M_1 and the jth row of M_2.

Construction 4.2: Direct product of measurement matrices.

- *Given:* Integer parameters n, m', d, u.
- *Output:* An $m \times n$ Boolean matrix M, where $m := m'\lceil \log(d/u) \rceil$.
- *Construction:* Let $r := \lceil \log(d/u) \rceil$. Index the rows of M by $[r] \times [m']$. Sample the (i, j)th row of M independently from a $(u + 1)$-wise independent distribution on n bit vectors, where each individual bit has probability $1/(2^{i+2}u)$ of being 1.

Construction 4.3: Probabilistic construction of regular and disjunct matrices.

As a corollary it follows that, when M_1 is a $(d-1, e_1; u-1)$-regular and M_2 is a (d, e_2)-disjunct matrix, the product $M := M_1 \odot M_2$ will distinguish between any two distinct d-sparse vectors (of weight at least u) in at least $(e_1 + 1)(e_2 + 1)$ positions of the measurement outcomes. This combined with Lemma 4.27 would imply that M is, in particular, $(\lfloor d/2 \rfloor, (e_1+1)(e_2+1)-1; u)$-disjunct. However, using a direct argument similar to the above lemma it is possible to obtain a slightly better result, given by Lemma 4.29 (the proof follows the same line of argument as that of Lemma 4.28 and is thus omitted).

Lemma 4.29. *Suppose that M_1 is a $(d, e_1; u-1)$-regular and M_2 is a $(2d, e_2)$-disjunct matrix. Then $M_1 \odot M_2$ is a $(d, (e_1+1)(e_2+1)-1; u)$-disjunct matrix.* □

As another particular example, we remark that the resilient measurement matrices that we constructed in Section 4.2.2 for the ordinary group testing model can be combined with regular matrices to offer the same qualities (i.e., approximation of sparse vectors in highly noisy settings) in the threshold model. In the same way, numerous existing results in group testing can be ported to the threshold model by using Lemma 4.28 (e.g., constructions of measurement matrices suitable for trivial two-stage schemes; cf. [29]).

4.3.3.2 Constructions

In this section, we obtain several constructions of regular and disjunct matrices. Our first construction, described in Construction 4.3, is a randomness-efficient probabilistic construction that can be analyzed using standard techniques from the probabilistic method. The bounds obtained by this construction are given by Lemma 4.30 below. The amount of random bits required by this construction is polynomially bounded in d and $\log n$, which is significantly smaller than it would be had we picked the entries of M fully independently.

Lemma 4.30. *For every $p \in [0,1)$ and integer parameter $u > 0$, Construction 4.3 with[17] $m' = O_u(d\log(n/d)/(1-p)^2)$ (resp., $m' = O_u(d^2\log(n/d)/(1-p)^2)$) outputs a $(d, \Omega_u(pm'); u)$-regular (resp., $(d, \Omega_u(pm'/d); u)$-disjunct) matrix with probability $1 - o(1)$.*

Proof. We show the claim for regular matrices, the proof for disjunct matrices is similar. Consider any particular choice of a critical set $S \subseteq [n]$ and a zero set $Z \subseteq [n]$ such that $u \le |S| \le d$ and $|Z| \le |S|$. Choose an integer i so that $2^{i-1}u \le |S| \le 2^i u$, and take any $j \in [m']$. Denote the (i,j)th row of M by the random variable $w \in \{0,1\}^n$, and by q the "success" probability that $w|_S$ has weight exactly u and $w|_Z$ is all zeros. For an integer $\ell > 0$, we will use the shorthand 1^ℓ (resp., 0^ℓ) for the all-ones (resp., all-zeros) vector of length ℓ. We have

$$\begin{aligned}
q &= \sum_{\substack{R\subseteq[S]\\|R|=u}} \Pr[(w|_R) = 1^u \wedge (w|_{Z\cup(S\setminus R)}) = 0^{|S|+|Z|-u}] \\
&= \sum_R \Pr[(w|_R) = 1^u] \cdot \Pr[(w|_{Z\cup(S\setminus R)}) = 0^{|S|+|Z|-u} \mid (w|_R) = 1^u] \\
&\overset{(a)}{=} \sum_R (1/(2^{i+2}u))^u \cdot (1 - \Pr[(w|_{Z\cup(S\setminus R)}) \neq 0^{|S|+|Z|-u} \mid (w|_R) = 1^u]) \\
&\overset{(b)}{\ge} \sum_R (1/(2^{i+2}u))^u \cdot (1 - (|S|+|Z|-u)/(2^{i+2}u)) \\
&\overset{(c)}{\ge} \frac{1}{2}\binom{|S|}{u}(1/(2^{i+2}u))^u \ge \frac{1}{2}\left(\frac{|S|}{u}\right)^u \cdot (1/(2^{i+2}u))^u \ge \frac{1}{2^{3u+1}\cdot u^u} =: c,
\end{aligned}$$

where (a) and (b) use the fact that the entries of w are $(u+1)$-wise independent, and (b) uses an additional union bound. Moreover, in (c) the binomial term counts the number of possibilities for the set R. Note that the lower bound $c > 0$ obtained at the end is a constant that only depends on u. Now, let $e := m'pq$, and observe that the expected number of "successful" rows is $m'q$. Using Chernoff bounds, and independence of the rows, the probability that there are at most e rows (among $(i,1),\ldots,(i,m')$) whose restriction to S and Z has weights u and 0, respectively, becomes upper bounded by

$$\exp(-(m'q-e)^2/(2m'q)) = \exp(-(1-p)^2 m'q/2) \le \exp(-(1-p)^2 m'c/2).$$

[17] The subscript in $O_u(\cdot)$ and $\Omega_u(\cdot)$ implies that the hidden constant in the asymptotic notation is allowed to depend on u.

- *Given:* A strong lossless (k, ϵ)-condenser $f\colon \{0,1\}^{\tilde{n}} \times \{0,1\}^t \to \{0,1\}^{\tilde{\ell}}$, integer parameter $u \geq 1$ and real parameter $p \in [0,1)$ such that $\epsilon < (1-p)/16$,
- *Output:* An $m \times n$ Boolean matrix M, where $n := 2^{\tilde{n}}$ and $m = 2^{t+k} O_u(2^{u(\tilde{\ell}-k)})$.
- *Construction:* Let $G_1 = (\{0,1\}^{\tilde{\ell}}, \{0,1\}^k, E_1)$ be any bipartite biregular graph with left vertex set $\{0,1\}^{\tilde{\ell}}$, right vertex set $\{0,1\}^k$, left degree $d_\ell := 8u$, and right degree $d_r := 8u2^{\tilde{\ell}-k}$. Replace each right vertex v of G_1 with $\binom{d_r}{u}$ vertices, one for each subset of size u of the vertices on the neighborhood of v, and connect them to the corresponding subsets. Denote the resulting graph by $G_2 = (\{0,1\}^{\tilde{\ell}}, V_2, E_2)$, where $|V_2| = 2^k\binom{d_r}{u}$. Define the bipartite graph $G_3 = (\{0,1\}^n, V_3, E_3)$, where $V_3 := \{0,1\}^t \times V_2$, as follows: Each left vertex $x \in \{0,1\}^n$ is connected to $(y, \Gamma_2(f(x,y)))$, for each $y \in \{0,1\}^t$, where $\Gamma_2(\cdot)$ denotes the neighborhood function of G_2 (i.e., $\Gamma_2(v)$ denotes the set of vertices adjacent to v in G_2). The output matrix M is the bipartite adjacency matrix of G_3.

Construction 4.4: A building block for construction of regular matrices.

Now take a union bound on all the choices of S and Z to conclude that the probability that the resulting matrix is not $(d, e; u)$-regular is at most

$$\left(\sum_{s=u}^{d}\binom{n}{s}\sum_{z=0}^{s}\binom{n-s}{z}\right)\exp(-(1-p)^2m'c/2)$$
$$\leq d^2\binom{n}{d}^2\exp(-(1-p)^2m'c/2),$$

which can be made $o(1)$ by choosing $m' = O_u(d\log(n/d)/(1-p)^2)$. □

Now we turn to a construction of regular matrices using strong lossless condensers. Details of the construction are described in Construction 4.5 that assumes a family of lossless condensers with different entropy requirements[18], and in turn, uses Construction 4.4 as a building block.

The following theorem analyzes the obtained parameters without specifying any particular choice for the underlying family of condensers.

[18] We have assumed that all the functions in the family have the same seed length t. If this is not the case, one can trivially set t to be the largest seed length in the family.

- *Given:* Integer parameters $d \geq u \geq 1$, real parameter $p \in [0, 1)$, and a family $f_0, \ldots, f_r$ of strong lossless condensers, where $r := \lceil \log(d/u') \rceil$ and u' is the smallest power of two such that $u' \geq u$. Each $f_i \colon \{0,1\}^{\tilde{n}} \times \{0,1\}^t \to \{0,1\}^{\tilde{\ell}(i)}$ is assumed to be a strong lossless $(k(i), \epsilon)$-condenser, where $k(i) := \log u' + i + 1$ and $\epsilon < (1-p)/16$.
- *Output:* An $m \times n$ Boolean matrix M, where $n := 2^{\tilde{n}}$ and $m = 2^t d \sum_{i=0}^{r} O_u(2^{u(\tilde{\ell}(i)-k(i))})$.
- *Construction:* For each $i \in \{0, \ldots, r\}$, denote by M_i the output matrix of Construction 4.4 when instantiated with f_i as the underlying condenser, and by m_i its number of rows. Define $r_i := 2^{r-i}$ and let M_i' denote the matrix obtained from M_i by repeating each row r_i times. Construct the output matrix M by stacking $M_0', \ldots, M_r'$ on top of one another.

Construction 4.5: Regular matrices from strong lossless condensers.

Theorem 4.31. *The $m \times n$ matrix M output by Construction 4.5 is $(d, p\gamma 2^t; u)$-regular, where $\gamma = \max\{1, \Omega_u(d \cdot \min\{2^{k(i)-\tilde{\ell}(i)} : i = 0, \ldots, r\})\}$.*

Proof. As a first step, we verify the upper bound on the number of measurements m. Each matrix M_i has $m_i = 2^{t+k(i)} O_u(2^{u(\tilde{\ell}(i)-k(i))})$ rows, and M_i' has $m_i r_i$ rows, where $r_i = 2^{r-i}$. Therefore, the number of rows of M is

$$\sum_{i=0}^{r} r_i m_i = \sum_{i=0}^{r} 2^{t+\log u'+r+1} m_i = 2^t d \sum_{i=0}^{r} O_u(2^{u(\tilde{\ell}(i)-k(i))}).$$

Let $S, Z \subseteq \{0,1\}^{\tilde{n}}$ respectively denote any choice of a critical set and zero set of size at most d, where $|Z| \leq |S|$, and choose an integer $i \geq 0$ so that $2^{i-1} u' \leq |S| \leq 2^i u'$. Arbitrarily grow the two sets S and Z to possibly larger, and disjoint, sets $S' \supseteq S$ and $Z' \supseteq Z$ such that $|S'| = |Z'| = 2^i u'$ (for simplicity we have assumed that $d \leq n/2$). Our goal is to show that there are "many" rows of the matrix M_i (in Construction 4.5) that u-satisfy S and Z.

Let $k := k(i) = \log u' + i + 1$, $\tilde{\ell} := \tilde{\ell}(i)$, and denote by G_1, G_2, G_3 the bipartite graphs used by the instantiation of Construction 4.4 that outputs M_i. Thus we need to show that "many" right vertices of G_3 are each connected to exactly u of the vertices in S and none of those in Z.

Consider the uniform distribution $\mathcal{X}$ on the set $S' \cup Z'$, which has min-entropy $\log u' + i + 1$. By an averaging argument, since the condenser f_i

is strong, for more than a p fraction of the choices of the seed $y \in \{0,1\}^t$ (call them *good seeds*), the distribution $\mathcal{Z}_y := f_i(\mathcal{X}, y)$ is $\epsilon/(1-p)$-close (in particular, 1/16-close) to a distribution with min-entropy $\log u' + i + 1$.

Fix any good seed $y \in \{0,1\}^t$. Let $G = (\{0,1\}^{\tilde{n}}, \{0,1\}^{\tilde{\ell}}, E)$ denote a bipartite graph representation of f_i, where each left vertex $x \in \{0,1\}^{\tilde{n}}$ is connected to $f_i(x,y)$ on the right. Denote by $\Gamma_y(S' \cup Z')$ the right vertices of G corresponding to the neighborhood of the set of left vertices picked by $S' \cup Z'$. Note that $\Gamma_y(S' \cup Z') = \mathsf{supp}(\mathcal{Z}_y)$. Using Proposition 2.14 in the appendix, we see that since $\mathcal{Z}_y$ is 1/16-close to having min-entropy $\log(|S' \cup Z'|)$, there are at least $(7/8)|S' \cup Z'|$ vertices in $\Gamma(S' \cup Z')$ that are each connected to exactly one left vertex in $S' \cup Z'$. Since $|S| \geq |S' \cup Z'|/4$, this implies that at least $|S' \cup Z'|/8$ vertices in $\Gamma(S' \cup Z')$ (call them Γ'_y) are connected to exactly one left vertex in S and no other vertex in $S' \cup Z'$. In particular we get that $|\Gamma'_y| \geq 2^{k-3}$.

Now, in G_1, let T_y be the set of left vertices corresponding to Γ'_y (regarding the left vertices of G_1 in one-to-one correspondence with the right vertices of G). The number of edges going out of T_y in G_1 is $d_\ell |T_y| \geq u 2^k$. Therefore, as the number of the right vertices of G_1 is 2^k, there must be at least one right vertex that is connected to at least u vertices in T_y. Moreover, a counting argument shows that the number of right vertices connected to at least u vertices in T_y is also at least $2^{k-\tilde{\ell}} 2^k/(10u)$.

Observe that in construction of G_2 from G_1, any right vertex of G_1 is replicated $\binom{d_r}{u}$ times, one for each u-subset of its neighbors. Therefore, for a right vertex of G_1 that is connected to *at least* u left vertices in T_y, one or more of its copies in G_2 must be connected to *exactly* u vertex in T_y (among the left vertices of G_2) and no other vertex (since the right degree of G_2 is equal to u).

Define $\gamma' := \max\{1, 2^{k-\tilde{\ell}} 2^k/(10u)\}$. From the previous argument we know that, looking at T_y as a set of left vertices of G_2, there are at least γ' right vertices on the neighborhood of T_y in G_2 that are connected to exactly u of the vertices in T_y and none of the left vertices outside T_y. Letting v_y be any such vertex, this implies that the vertex $(y, v_y) \in V_3$ on the right part of G_3 is connected to exactly u of the vertices in S, and none of the vertices in Z. Since the argument holds for every good seed y, the number of such vertices is at least the number of good seeds, which is more than $p\gamma' 2^t$. Since the rows of the matrix m_i are repeated $r_i = 2^{r-i}$ times in M, we conclude that M has at least $p\gamma' 2^{t+r-i} \geq p\gamma 2^t$ rows that u-satisfy S and Z, and the claim follows. □

Instantiations

We now instantiate the result obtained in Theorem 4.31 by various choices of the family of lossless condensers. The crucial factors that influence the number of measurements are the seed length and the output length of the condenser. In particular, we will consider optimal lossless condensers (with

parameters achieved by random functions), zig-zag based construction of Theorem 4.19, and the coding-theoretic construction of Guruswami et al., quoted in Theorem 2.22. The results are summarized in the following theorem.

Theorem 4.32. *Let $u > 0$ be fixed, and $p \in [0, 1)$ be a real parameter. Then for integer parameters $d, n \in \mathbb{N}$ where $u \leq d \leq n$,*

1. *Using an optimal lossless condenser in Construction 4.5 results in an $m_1 \times n$ matrix M_1 that is $(d, e_1; u)$-regular, where*

$$m_1 = O(d(\log n)(\log d)/(1-p)^{u+1})$$

and $e_1 = \Omega(pd \log n)$,

2. *Using the lossless condenser of Theorem 4.19 in Construction 4.5 results in an $m_2 \times n$ matrix M_2 that is $(d, e_2; u)$-regular, where*

$$m_2 = O(T_2 d(\log d)/(1-p)^u)$$

for some

$$T_2 = \exp(O(\log^3((\log n)/(1-p)))) = \mathsf{quasipoly}(\log n),$$

and $e_2 = \Omega(pdT_2(1-p))$.

3. *Let $\beta > 0$ be any fixed constant. Then Construction 4.5 can be instantiated using the lossless condenser of Theorem 2.22 so that we obtain an $m_3 \times n$ matrix M_3 that is $(d, e_3; u)$-regular, where*

$$m_3 = O(T_3^{1+u} d^{1+\beta}(\log d))$$

for

$$T_3 := ((\log n)(\log d)/(1-p))^{1+u/\beta} = \mathsf{poly}(\log n, \log d),$$

and $e_3 = \Omega(p \max\{T_3, d^{1-\beta/u}\})$.

Proof. First we show the claim for M_1. In this case, we take each f_i in Construction 4.5 to be an optimal lossless condenser satisfying the bounds obtained in[19] [23]. Thus we have that $2^t = O(\tilde{n}/\epsilon) = O(\log n/\epsilon)$, and for every $i = 0, \ldots, r$, we have $2^{\tilde{\ell}(i)-k(i)} = O(1/\epsilon)$, where $\epsilon = O(1-p)$. Now we apply Theorem 4.31 to obtain the desired bounds (and in particular, $\gamma = \Omega(\epsilon d)$).

Similarly, for the construction of M_2 we set up each f_i using the explicit construction of condensers in Theorem 4.19 for min-entropy $k(i)$. In this case, the maximum required seed length is $t = O(\log^3(\tilde{n}/\epsilon))$, and we let

$$T_2 := 2^t = \exp(O(\log^3((\log n)/(1-p)))).$$

[19] This result is similar in spirit to the probabilistic argument used in [122] for showing the existence of good extractors.

Moreover, for every $i = 0, \ldots, r$, we have $2^{\tilde{\ell}(i)-k(i)} = O(1/\epsilon)$. Plugging these parameters in Theorem 4.31 gives $\gamma = \Omega(\epsilon d)$ and the bounds on m_2 and e_2 follow.

Finally, for M_3 we use Theorem 2.22 with $\alpha := \beta/u$. Thus the maximum seed length becomes

$$t = (1 + u/\beta) \log(\tilde{n}(\log d)/(1-p)) + O(1),$$

and for every $i = 0, \ldots, r$, we have $\tilde{\ell}(i) - k(i) = O(t + \beta(\log d)/u)$. Clearly, $T_3 = \Theta(2^t)$, and thus (using Theorem 4.31) the number of measurements becomes $m_3 = T^{1+u} d^{1+\beta} (\log d)$. Moreover, we get

$$\gamma = \max\{1, \Omega(d^{1-\beta/u}/T)\},$$

which gives

$$e_3 = \Omega(pT\gamma) = p \max\{T, d^{1-\beta/u}\},$$

as claimed. □

By combining this result with Lemma 4.29 using any explicit construction of classical disjunct matrices, we will obtain $(d, e; u)$-disjunct matrices that can be used in the threshold model with any fixed threshold, sparsity d, and error tolerance $\lfloor e/2 \rfloor$.

In particular, using the coding-theoretic explicit construction of nearly optimal classical disjunct matrices (see Table 4.2), we obtain $(d, e; u)$-disjunct matrices with

$$m = O(m' d^2 (\log n)/(1-p)^2)$$

rows and error tolerance

$$e = \Omega(e' p d (\log n)/(1-p)),$$

where m' and e' are respectively the number of rows and error tolerance of any of the regular matrices obtained in Theorem 4.32.

We note that in all cases, the final dependence on the sparsity parameter d is, roughly, $O(d^3)$ which has an exponent independent of the threshold u. Table 4.3 summarizes the obtained parameters for the general case (with arbitrary gaps). We see that, when d is not negligibly small (e.g., $d = n^{1/10}$), the bounds obtained by our explicit constructions are significantly better than those offered by strongly disjunct matrices (as in Table 4.2).

4.3.3.3 The Case with Positive Gaps

In preceding sections we have focused on the case where $g = 0$. However, we observe that all the techniques that we have developed so far can be extended to the positive-gap case in a straightforward way. The main observations are as follows.

Table 4.3: Summary of the parameters achieved by various threshold testing schemes. The noise parameter $p \in [0,1)$ is arbitrary, and thresholds $\ell, u = \ell + g$ are fixed constants. "Exp" and "Rnd" respectively indicate explicit and randomized constructions.

Number of rows	Tolerable errors	Remarks
$O(d^{g+2}\frac{(\log d)\log(n/d)}{(1-p)^2})$	$\Omega(pd\frac{\log(n/d)}{(1-p)^2})$	Rnd: Construction 4.3.
$O(d^{g+3}\frac{(\log d)\log^2 n}{(1-p)^2})$	$\Omega(pd^2\frac{\log^2 n}{(1-p)^2})$	Constructions 4.5 and 4.2 combined, assuming optimal condensers and strongly disjunct matrices.
$O(d^{g+3}\frac{(\log d)T_2\log n}{(1-p)^{g+2}})$	$\Omega(pd^2\frac{T_2\log n}{1-p})$	Exp ($\star$)
$O(d^{g+3+\beta}\frac{T_3^\ell\log n}{(1-p)^{g+2}})$	$\Omega(pd^{2-\beta}\frac{\log n}{1-p})$	Exp ($\star\star$)
$\Omega(d^{g+2}\log_d n + ed^{g+1})$	e	Lower bound (see Section 4.3.3.3).

($\star$) Constructions 4.5 and 4.2 combined using Theorem 4.19 and [120], where $T_2 = \exp(O(\log^3 \log n)) = \mathsf{quasipoly}(\log n)$.

($\star\star$) Constructions 4.5 and 4.2 combined using Theorem 2.22 and [120], where $\beta > 0$ is any arbitrary constant and $T_3 = ((\log n)(\log d))^{1+u/\beta} = \mathsf{poly}(\log n, \log d)$.

---

1. Definition 4.26 can be adapted to allow more than a single distinguished column in disjunct matrices. In particular, in general we may require the matrix M to have more than e rows that u-satisfy every choice of a critical set S, a zero set Z, and any $g+1$ designated columns $D \subseteq S$ (at which all entries of the corresponding rows must be 1). Denote this generalized notion by $(d, e; u, g)$-disjunct matrices. It is straightforward to extend the arguments of Lemma 4.27 to show that the generalized notion of $(d, e; u, g)$-disjunct matrices is necessary and sufficient to capture non-adaptive threshold group testing with upper threshold u and gap g.

2. Lemma 4.30 can be generalized to show that Construction 4.3 (with probability $1-o(1)$) results in a $(d, \Omega_u(pd\log(n/d)/(1-p)^2); u, g)$-disjunct matrix if the number of measurements is increased by a factor $O(d^g)$.

3. Lemma 4.28 can be extended to positive gaps, by taking M_1 as a $(d-1, e_1; \ell-1)$-regular matrix, provided that, for every $y \in M_2[x]_{1,g+1}$ and $y' \in M_2[x']_{1,g+1}$, we have $|\mathsf{supp}(y) \setminus \mathsf{supp}(y')| \geq e_2$. In particular this

is the case if M_2 is strongly $(d, e_2 - 1; g + 1)$-disjunct[20]. Similarly for Lemma 4.29, M_2 must be taken as a strongly $(2d, e_2; g+1)$-disjunct matrix. Consequently, using the coding-theoretic construction of strongly disjunct matrices described in Section 4.3.2, our explicit constructions of $(d, e; u)$-disjunct matrices can be extended to the gap model at the cost of a factor $O(d^g)$ increase in the number of measurements (as summarized in Table 4.3).

4. Observe that a $(d, e; u, g)$-disjunct matrix is in particular, strongly $(d - g, e; g + 1)$-disjunct and thus, the lower bound $\Omega(d^{g+2} \log_d n + ed^{g+1})$ on the number of rows of strongly disjunct matrices applies to them as well.

4.4 Notes

The notion of d-disjunct matrices is also known in certain equivalent forms; e.g., *d-superimposed codes*, *d-separable matrices*, or *d-cover-free families* (cf. [50]). The special case of Definition 4.7 corresponding to $(0, 0, e'_0, 0)$-resilient matrices is related to the notion of *selectors* in [43] and *resolvable matrices* in [56]. Lemma 4.10 is similar in spirit to the lower bound obtained in [43] for the size of selectors.

The notion of strongly disjunct matrices, in its general form, has been studied in the literature under different names and equivalent formulations, e.g., superimposed (u, d)-designs/codes and (u, d) cover-free families (see [26, 28, 53, 91, 144, 145] and the references therein).

4.A Some Technical Details

For a positive integer $c > 1$, define a c-hypergraph as a tuple (V, E), where V is the set of vertices and E is the set of hyperedges such that every $e \in E$ is a subset of V of size c. The degree of a vertex $v \in V$, denoted by $\mathsf{deg}(v)$, is the size of the set $\{e \in E \colon v \in E\}$. Note that $|E| \leq \binom{|V|}{c}$ and $\mathsf{deg}(v) \leq \binom{|V|}{c-1}$. The *density* of the hypergraph is given by $|E|/\binom{|V|}{c}$. A *vertex cover* on the hypergraph is a subset of vertices that contains at least one vertex from every hyperedge. A *matching* is a set of pairwise disjoint hyperedges. It is well known that any dense hypergraph must have a large matching. Below we reconstruct a proof of this claim.

Proposition 4.33. *Let H be a c-hypergraph such that every vertex cover of H has size at least k. Then H has a matching of size at least k/c.*

Proof. Let M be a maximal matching of H, i.e., a matching that cannot be extended by adding further hyperedges. Let C be the set of all vertices that

[20] Here we are also considering the unavoidable assumption that $\max\{|\mathsf{supp}(x) \setminus \mathsf{supp}(x')|, |\mathsf{supp}(x') \setminus \mathsf{supp}(x)|\} > g$.

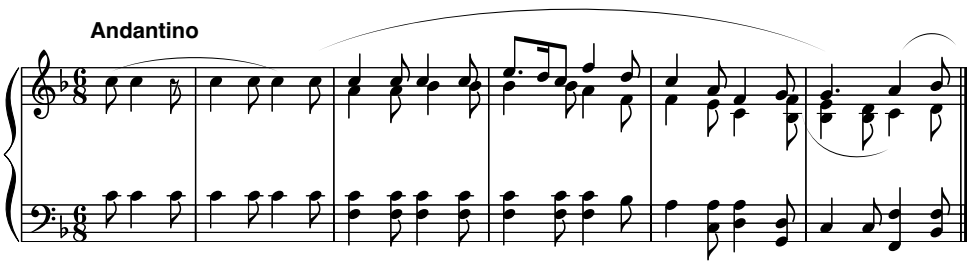
participate in hyperedges of M. Then C has to be a vertex cover, as otherwise one could add an uncovered hyperedge to M and violate maximality of M. Hence, $c|M| = |C| \geq k$, and the claim follows. □

Lemma 4.34. *Let $H = (V, E)$ be a c-hypergraph with density at least $\epsilon > 0$. Then H has a matching of size at least $\frac{\epsilon}{c^2}(|V| - c + 1)$.*

Proof. For every subset $S \subseteq V$ of size c, denote by $\mathbb{1}(S)$ the indicator value of S being in E. Let C be any vertex cover of H. Denote by $\mathcal{S}$ the set of all subsets of V of size c. Then we have

$$\epsilon\binom{|V|}{c} \leq \sum_{S \in \mathcal{S}} \mathbb{1}(S) \leq \sum_{v \in C} \mathsf{deg}(v) \leq |C|\binom{|V|}{c-1}.$$

Hence, $|C| \geq \epsilon(n - c + 1)/c$, and the claim follows using Proposition 4.33. □



Frédéric Chopin (1810–1849): Ballade Op. 38 No. 2 in F major.

"How is an error possible in mathematics?"
— Henri Poincaré

Chapter 5

Capacity Achieving Codes

One of the basic goals of coding theory is coming up with efficient constructions of error-correcting codes that allow reliable transmission of information over discrete communication channels. Already in the seminal work of Shannon [136], the notion of *channel capacity* was introduced which is a characteristic of the communication channel that determines the maximum rate at which reliable transmission of information (i.e., with vanishing error probability) is possible. However, Shannon's result did not focus on the *feasibility* of the underlying code and mainly concerned with the existence of reliable, albeit possibly complex, coding schemes. Here feasibility can refer to a combination of several criteria, including: succinct description of the code and its efficient computability, the existence of an efficient encoder and an efficient decoder, the error probability, and the set of message lengths for which the code is defined.

Besides heuristic attempts, there is a large body of rigorous work in the literature on coding theory with the aim of designing feasible capacity approaching codes for various discrete channels, most notably, the natural and fundamental cases of the binary erasure channel (BEC) and binary symmetric channel (BSC). Some notable examples in "modern coding" include Turbo codes and sparse graph codes (e.g., LDPC codes and Fountain codes, cf. [13, 125, 137]). These classes of codes are either known or strongly believed to contain capacity achieving ensembles for the erasure and symmetric channels.

While such codes are very appealing both theoretically and practically, and are in particular designed with efficient decoding in mind, in this area there still is a considerable gap between what we can prove and what is evidenced by practical results, mainly due to complex combinatorial structure of the code constructions. Moreover, almost all known code constructions in this area involve a considerable amount of randomness, which makes them prone to a possibility of design failure (e.g., choosing an "unfortunate" degree sequence

for an LDPC code). While the chance of such possibilities is typically small, in general there is no known efficient way to certify whether a particular outcome of the code construction is satisfactory. Thus, it is desirable to come up with constructions of provably capacity achieving code families that are explicit, i.e., are efficient and do not involve any randomness.

Explicit construction of capacity achieving codes was considered as early as the classic work of Forney [60], who showed that concatenated codes can achieve the capacity of various memoryless channels. In this construction, an outer MDS code is concatenated with an inner code with small block length that can be found in reasonable time by brute force search. An important subsequent work by Justesen [87] (that was originally aimed for explicit construction of asymptotically good codes) shows that it is possible to eliminate the brute force search by varying the inner code used for encoding different symbols of the outer encoding, provided that the ensemble of inner codes contains a large fraction of capacity achieving codes.

Recently, Arikan [7] gave a framework for deterministic construction of capacity achieving codes for discrete memoryless channels (DMCs) with binary input that are equipped with efficient encoders and decoders and attain slightly worse than exponentially small error probability. These codes are defined for every block length that is a power of two, which might be considered a restrictive requirement. Moreover, the construction is currently explicit (in the sense of polynomial-time computability of the code description) only for the special case of BEC and requires exponential time otherwise.

In this chapter, we revisit the concatenation scheme of Justesen and give new constructions of the underlying ensemble of the inner codes. The code ensemble used in Justesen's original construction is attributed to Wozencraft. Other ensembles that are known to be useful in this scheme include the ensemble of Goppa codes and shortened cyclic codes (see [127], Chapter 12). The number of codes in these ensembles is exponential in the block length and they achieve exponentially small error probability. These ensembles are also known to achieve the Gilbert-Varshamov bound, and owe their capacity achieving properties to the property that each nonzero vector belongs to a small number of the codes in the ensemble.

Here, we will use extractors and lossless condensers to construct much smaller ensembles with similar, random-like, properties. The quality of the underlying extractor or condenser determines the quality of the resulting code ensemble. In particular, the size of the code ensemble, the decoding error and proximity to the channel capacity are determined by the *seed length*, the *error*, and the *output length* of the extractor or condenser being used.

As a concrete example, we will instantiate our construction with appropriate choices of the underlying condenser (or extractor) and obtain, for every block length n, a capacity achieving ensemble of size 2^n that attains exponentially small error probability for both erasure and symmetric channels (as well as the broader range of channels described above), and an ensemble of

quasipolynomial[1] size $2^{O(\log^3 n)}$ that attains the capacity of BEC. Using nearly optimal extractors and condensers that require logarithmic seed lengths, it is possible to obtain polynomially small capacity achieving ensembles for any block length.

Finally, we apply our constructions to Justesen's concatenation scheme to obtain an explicit construction of capacity-achieving codes for both BEC and BSC that attain exponentially small error, as in the original construction of Forney. Moreover, the running time of the encoder is almost linear in the block length, and decoding takes almost linear time for BEC and almost quadratic time for BSC. Using our quasipolynomial-sized ensemble as the inner code, we are able to construct a fully explicit code for BEC that is defined and capacity achieving for every choice of the message length.

5.1 Discrete Communication Channels

A *discrete communication channel* is a randomized process that takes a potentially infinite stream of symbols $X_0, X_1, \ldots$ from an *input alphabet* Σ and outputs an infinite stream $Y_0, Y_1, \ldots$ from an *output alphabet* Γ. The indices intuitively represent the *time*, and each output symbol is only determined from what channel has observed in the past. More precisely, given $X_0, \ldots, X_t$, the output symbol Y_t must be independent of $X_{t+1}, X_{t+2}, \ldots$. Here we will concentrate on finite input and finite output channels, that is, the alphabets Σ and Γ are finite. In this case, the conditional distribution $p(Y_t|X_t)$ of each output symbol Y_t given the input symbol X_t can be written as a stochastic $|\Sigma| \times |\Gamma|$ *transition matrix*, where each row is a probability distribution.

Of particular interest is a *memoryless* channel, which is intuitively "oblivious" of the past. In this case, the transition matrix is independent of the time instance. That is, we have $p(Y_t|X_t) = p(Y_0|X_0)$ for every t. When the rows of the transition matrix are permutations of one another and so is the case for the columns, the channel is called *symmetric*. For example, the channel defined by

$$p(Y|X) = \begin{pmatrix} 0.4 & 0.1 & 0.5 \\ 0.5 & 0.4 & 0.1 \\ 0.1 & 0.5 & 0.4 \end{pmatrix}$$

is symmetric. Intuitively, a symmetric channel does not "read" the input sequence. An important class of symmetric channels is defined by *additive noise*. In an additive noise channel, the input and output alphabets are the same finite field $\mathbb{F}_q$ and each output symbol Y_t is obtained from X_t using

$$Y_t = X_t + Z_t,$$

[1] A quantity $f(n)$ is said to be quasipolynomial in n (denoted by $f(n) = \mathsf{quasipoly}(n)$) if $f(n) = 2^{(\log n)^{O(1)}}$.

where the addition is over $\mathbb{F}_q$ and the channel noise $Z_t \in \mathbb{F}_q$ is chosen independently of the input sequence[2]. Typically Z_t is also independent of time t, in which case we get a memoryless additive noise channel. For a noise distribution $\mathcal{Z}$, we denote the memoryless additive noise channel over the input (as well as output) alphabet Σ by $\mathsf{SC}(\Sigma, \mathcal{Z})$.

Note that the notion of additive noise channels can be extended to the case where the input and alphabet sets are vector spaces $\mathbb{F}_q^n$, and the noise distribution is a probability distribution over $\mathbb{F}_q^n$. By considering an isomorphism between $\mathbb{F}_q^n$ and the field extension $\mathbb{F}_{q^n}$, such a channel is essentially an additive noise channel $\mathsf{SC}(\mathbb{F}_{q^n}, \mathcal{Z})$, where $\mathcal{Z}$ is a noise distribution over $\mathbb{F}_{q^n}$. On the other hand, the channel $\mathsf{SC}(\mathbb{F}_{q^n}, \mathcal{Z})$ can be regarded as a "block-wise memoryless" channel over the alphabet $\mathbb{F}_q$. Namely, in a natural way, each channel use over the alphabet $\mathbb{F}_{q^n}$ can be regarded as n subsequent uses of a channel over the alphabet $\mathbb{F}_q$. When regarding the channel over $\mathbb{F}_q$, it does not necessarily remain memoryless since the additive noise distribution $\mathcal{Z}$ can be an arbitrary distribution over $\mathbb{F}_{q^n}$ and is not necessarily expressible as a product distribution over $\mathbb{F}_q$. However, the noise distribution of blocks of n subsequent channel uses are independent from one another and form a product distribution (since the original channel $\mathsf{SC}(\mathbb{F}_{q^n}, \mathcal{Z})$ is memoryless over $\mathbb{F}_{q^n}$). Often by choosing larger and larger values of n and letting n grow to infinity, it is possible to obtain good approximations of a non-memoryless additive noise channel using memoryless additive noise channels over large alphabets.

An important additive noise channel is the *q-ary symmetric channel*, which is defined by a (typically small) noise parameter $p \in [0, 1)$. For this channel, the noise distribution $\mathcal{Z}$ has a probability mass $1 - p$ on zero, and $p/(q-1)$ on every nonzero alphabet letter. A fundamental special case is the *binary symmetric channel* (BSC), which corresponds to the case $q = 2$ and is denoted by $\mathsf{BSC}(p)$.

Another fundamentally important channel is the *binary erasure channel*. The input alphabet for this channel is $\{0, 1\}$ and the output alphabet is the set $\{0, 1, ?\}$. The transition is characterized by an *erasure probability* $p \in [0, 1)$. A transmitted symbol is output intact by the channel with probability $1 - p$. However, with probability p, a special *erasure symbol* "?" is delivered by the channel. The behavior of the binary symmetric channel $\mathsf{BSC}(p)$ and binary erasure channel $\mathsf{BEC}(p)$ is schematically described by Figure 5.1.

A *channel encoder* $\mathcal{E}$ for a channel $\mathcal{C}$ with input alphabet Σ and output alphabet Γ is a mapping $\mathcal{C}\colon \{0,1\}^k \to \Sigma^n$. A *channel decoder*, on the other hand, is a mapping $\mathcal{D}\colon \Gamma^n \to \{0,1\}^k$. A channel encoder and a channel decoder collectively describe a *channel code*. Note that the image of the encoder mapping defines a block code of length n over the alphabet Σ. The parameter

[2] In fact, since we are only using the additive structure of $\mathbb{F}_q$, it can be replaced by any additive group, and in particular, the ring $\mathbb{Z}/q\mathbb{Z}$ for an arbitrary integer $q > 1$. This way, q does not need to be restricted to a prime power.

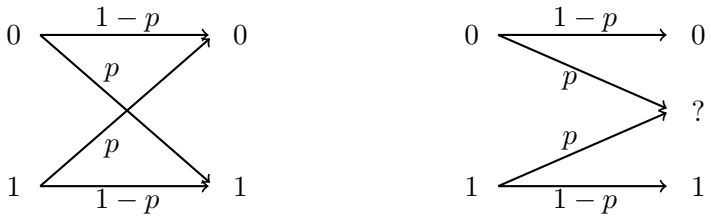


Figure 5.1: The binary symmetric channel (left) and binary erasure channel (right). On each graph, the left part corresponds to the input alphabet and the right part to the output alphabet. Conditional probability of each output symbol given an input symbol is shown by the labels on the corresponding arrows.

n defines the *block length* of the code. For a sequence $Y \in \Sigma^n$, denote by the random variable $\mathcal{C}(Y)$ a sequence $\hat{Y} \in \Gamma^n$ that is output by the channel, given the input Y.

Intuitively, a channel encoder adds sufficient redundancy to a given "message" $X \in \{0,1\}^k$ (that is without loss of generality modeled as a binary string of length k), resulting in an encoded sequence $Y \in \Sigma^n$ that can be fed into the channel. The channel manipulates the encoded sequence and delivers a sequence $\hat{Y} \in \Gamma^n$ to a recipient whose aim is to recover X. The recovery process is done by applying the channel decoder on the received sequence $\hat{Y}$. The transmission is successful when $\mathcal{D}(\hat{Y}) = X$. Since the channel behavior is not deterministic, there might be a nonzero probability, known as the *error probability*, that the transmission is unsuccessful. More precisely, the error probability of a channel code is defined as

$$p_e := \sup_{X \in \{0,1\}^k} \Pr[\mathcal{D}(\mathcal{C}(\mathcal{E}(X))) \neq X],$$

where the probability is taken over the randomness of $\mathcal{C}$. A schematic diagram of a simple communication system consisting of an encoder, point-to-point channel, and decoder is shown in Figure 5.2.

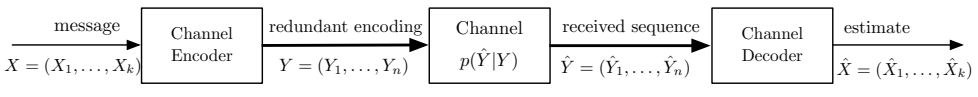


Figure 5.2: The schematic diagram of a point-to-point communication system. The stochastic behavior of the channel is captured by the conditional probability distribution $p(\hat{Y}|Y)$.

For linear codes over additive noise channels, it is often convenient to work with *syndrome decoders*. Consider a linear code with generator and parity check matrices G and H, respectively. The encoding of a message x (considered as a row vector) can thus be written as xG. Suppose that the encoded sequence is transmitted over an additive noise channel, which produces a noisy sequence $y := xG + z$, for a randomly chosen z according to the channel distribution. The receiver receives the sequence y and, without loss of generality, the decoder's task is to obtain an estimate of the noise realization z from y. Now, observe that

$$Hy^\top = HG^\top x^\top + Hz^\top = Hz^\top,$$

where the last equality is due to the orthogonality of the generator and parity check matrices. Therefore, $Hz^\top$ is available to the decoder and thus, in order to decode the received sequence, it suffices to obtain an estimate of the noise sequence z from the *syndrome* $Hz^\top$. A syndrome decoder is a function that, given the syndrome, outputs an estimate of the noise sequence (note that this is independent of the codeword being sent). The error probability of a syndrome decoder can be simply defined as the probability (over the noise randomness) that it obtains an incorrect estimate of the noise sequence. Obviously, the error probability of a syndrome decoder upper bounds the error probability of the channel code.

The *rate* of a channel code (in bits per channel use) is defined as the quantity k/n. We call a rate $r \geq 0$ *feasible* if for every $\epsilon > 0$, there is a channel code with rate r and error probability at most ϵ. The rate of a channel code describes its efficiency; the larger the rate, the more information can be transmitted through the channel in a given "time frame". A fundamental question is, given a channel $\mathcal{C}$, to find the largest possible rate at which reliable transmission is possible. In his fundamental work, Shannon [136] introduced the notion of *channel capacity* that answers this question. Shannon capacity can be defined using purely information-theoretic terminology. However, for the purposes of this chapter, it is more convenient to use the following, more "computational", definition which turns out to be equivalent to the original notion of Shannon capacity:

$$\mathsf{Cap}(\mathcal{C}) := \sup\{r \mid r \text{ is a feasible rate for the channel } \mathcal{C}\}.$$

Capacity of memoryless symmetric channels has a particularly nice form. Let $\mathcal{Z}$ denote the probability distribution defined by any of the rows of the transition matrix of a memoryless symmetric channel $\mathcal{C}$ with output alphabet Γ. Then, capacity of $\mathcal{C}$ is given by

$$\mathsf{Cap}(\mathcal{C}) = \log_2 |\Gamma| - H(\mathcal{Z}),$$

where $H(\cdot)$ denotes the Shannon entropy [40, Section 7.2]. In particular, capacity of the binary symmetric channel $\mathsf{BSC}(p)$ (in bits per channel use) is

equal to

$$1 - h(p) = 1 + p \log_2 p + (1-p) \log_2 (1-p).$$

Capacity of the binary erasure channel $\mathsf{BEC}(p)$ is moreover known to be $1-p$ [40, Section 7.1].

A *family* of channel codes of rate r is an infinite set of channel codes, such that for every (typically small) *rate loss* $\delta \in (0, r)$ and block length n, the family contains a code $\mathcal{C}(n, \delta)$ of length at least n and rate at least $r - \delta$. The family is called *explicit* if there is a deterministic algorithm that, given n and δ as parameters, computes the encoder function of the code $\mathcal{C}(n, \delta)$ in polynomial time in n. For linear channel codes, this is equivalent to computing a generator or parity check matrix of the code in polynomial time. If, additionally, the algorithm receives an auxiliary index $i \in [s]$, for a *size parameter* s depending on n and δ, we instead get an *ensemble* of size s of codes. An ensemble can be interpreted as a set of codes of length n and rate at least $r - \delta$ each, that contains a code for each possibility of the index i.

We call a family of codes *capacity achieving* for a channel $\mathcal{C}$ if the family is of rate $\mathsf{Cap}(\mathcal{C})$ and moreover, the code $\mathcal{C}(n, \delta)$ as described above can be chosen to have an arbitrarily small error probability for the channel $\mathcal{C}$. If the error probability decays exponentially with the block length n; i.e., $p_e = O(2^{-\gamma n})$, for a constant $\gamma > 0$ (possibly depending on the rate loss), then the family is said to achieve an *error exponent* γ. We call the family *capacity achieving for all lengths* if it is capacity achieving and moreover, there is an integer constant n_0 (depending only on the rate loss δ) such that for every $n \geq n_0$, the code $\mathcal{C}(n, \delta)$ can be chosen to have length exactly n.

5.2 Codes for the Binary Erasure Channel

Any code with minimum distance d can tolerate up to $d - 1$ erasures in the *worst case*[3]. Thus one way to ensure reliable communication over $\mathsf{BEC}(p)$ is to use binary codes with relative minimum distance of about p. However, known negative bounds on the rate-distance trade-off (e.g., the sphere packing and MRRW bounds) do not allow the rate of such codes to approach the capacity $1 - p$. However, by imposing the weaker requirement that *most* of the erasure patterns should be recoverable, it is possible to attain the capacity with a positive, but arbitrarily small, error probability (as guaranteed by the definition of capacity).

In this section, we consider a different relaxation that preserves the worst-case guarantee on the erasure patterns; namely we consider *ensembles* of linear codes with the property that *any* pattern of up to p erasures must be tolerable by all but a negligible fraction of the codes in the ensemble. This in particular allows us to construct ensembles in which all but a negligible fraction of the

[3] See Appendix A for a quick review of the basic notions in coding theory.

codes are capacity achieving for BEC. Note that as we are only considering linear codes, recoverability from a particular erasure pattern $S \subseteq [n]$ (where n is the block length) is a property of the code and independent of the encoded sequence.

Now we introduce two constructions, which employ strong, linear extractors and lossless condensers as their main ingredients. Throughout this section we denote by $f\colon \mathbb{F}_2^n \times \mathbb{F}_2^d \to \mathbb{F}_2^r$ a strong, linear, lossless condenser for min-entropy m and error ϵ and by $g\colon \mathbb{F}_2^n \times \mathbb{F}_2^{d'} \to \mathbb{F}_2^k$ a strong, linear extractor for min-entropy $n - m$ and error ϵ'. We assume that the errors ϵ and ϵ' are substantially small. Using this notation, we define the ensembles $\mathcal{F}$ and $\mathcal{G}$ as in Construction 5.1.

Obviously, the rate of each code in $\mathcal{F}$ is at least $1 - r/n$. Moreover, as g is a strong extractor we can assume without loss of generality that the rank of each G_u is exactly[4] k. Thus, each code in $\mathcal{G}$ has rate k/n. Lemma 5.2 below is our main tool in quantifying the erasure decoding capabilities of the two ensembles. Before stating the lemma, we mention a proposition showing that linear condensers applied on *affine sources* achieve either zero or large errors:

Proposition 5.1. *Suppose that a distribution $\mathcal{X}$ is uniformly supported on an affine k-dimensional subspace over $\mathbb{F}_q^n$. Consider a linear function $f\colon \mathbb{F}_q^n \to \mathbb{F}_q^m$, and define the distribution $\mathcal{Y}$ as $\mathcal{Y} := f(\mathcal{X})$. Suppose that, for some integer k and $\epsilon < 1/2$, $\mathcal{Y}$ is ϵ-close to having either min-entropy $m \log q$ or at least $k \log q$. Then, $\epsilon = 0$.*

Proof. By linearity, $\mathcal{Y}$ is uniformly supported on an affine subspace A of $\mathbb{F}_q^m$. Let $k' \leq m$ be the dimension of this subspace, and observe that $k' \leq k$.

[4] This causes no loss of generality since, if the rank of some G_u is not maximal, one of the k symbols output by the linear function $g(\cdot, u)$ would linearly depend on the others and thus, the function would fail to be an extractor for any source (so one can arbitrarily modify $g(\cdot, u)$ to have rank k without negatively affecting the parameters of the extractor g).

Ensemble $\mathcal{F}$: Define a code $\mathcal{C}_u$ for each seed $u \in \mathbb{F}_2^d$ as follows: Let H_u denote the $r \times n$ matrix that defines the linear function $f(\cdot, u)$, i.e., for each $x \in \mathbb{F}_2^n$, $H_u \cdot x = f(x, u)$. Then H_u is a parity check matrix for $\mathcal{C}_u$.

Ensemble $\mathcal{G}$: Define a code $\mathcal{C}'_u$ for each seed $u \in \mathbb{F}_2^{d'}$ as follows: Let G_u denote the $k \times n$ matrix that defines the linear function $g(\cdot, u)$. Then G_u is a generator matrix for $\mathcal{C}'_u$.

Construction 5.1: Ensembles $\mathcal{F}$ and $\mathcal{G}$ of error-correcting codes.

First, suppose that $\mathcal{Y}$ is ϵ-close to a distribution with min-entropy $m \log q$; i.e., the uniform distribution on $\mathbb{F}_q^m$. Now, the statistical distance between $\mathcal{Y}$ and the uniform distribution is, by definition,

$$\sum_{x \in A} (q^{-k'} - q^{-m}) = 1 - q^{k'-m} - 1.$$

Since $\epsilon < 1/4$, $q \geq 2$, and k' and m are integers, this implies that the distance is greater than $1/2$ (a contradiction) unless $k' = m$, in which case it becomes zero. Therefore, the output distribution is exactly uniform over $\mathbb{F}_q^m$.

Now consider the case where $\mathcal{Y}$ is ϵ-close to having min-entropy at least $k \log q$. Considering that $k' \leq k$, the definition of statistical distance implies that ϵ is at least

$$\sum_{x \in A} (q^{-k'} - q^{-k}) = 1 - q^{k'-k}.$$

Similarly as before, we get that $k' = k$, meaning that $\mathcal{Y}$ is precisely a distribution with min-entropy $k \log q$. □

Lemma 5.2. *Let $S \subseteq [n]$ be a set of size at most m. Then all but a 5ϵ fraction of the codes in $\mathcal{F}$ and all but a $5\epsilon'$ fraction of those in $\mathcal{G}$ can tolerate the erasure pattern defined by S.*

Proof. We prove the result for the ensemble $\mathcal{G}$. The argument for $\mathcal{F}$ is similar. Consider a probability distribution $\mathcal{S}$ on $\mathbb{F}_2^n$ that is uniform on the coordinates specified by $\bar{S} := [n] \setminus S$ and fixed to zeros elsewhere. Thus the min-entropy of $\mathcal{S}$ is at least $n - m$, and the distribution $(U, g(\mathcal{S}, U))$, where $U \sim \mathcal{U}_{d'}$, is ϵ'-close to $\mathcal{U}_{d'+k}$.

By Corollary 2.13, for all but a $5\epsilon'$ fraction of the choices of $u \in \mathbb{F}_2^{d'}$, the distribution of $g(\mathcal{S}, u)$ is $(1/5)$-close to $\mathcal{U}_k$. Fix such a u. By Proposition 5.1, the distribution of $g(\mathcal{S}, u)$ must in fact be exactly uniform. Thus, the $k \times m$ submatrix of G_u consisting of the columns picked by $\bar{S}$ must have rank k, which implies that for every $x \in \mathbb{F}_2^k$, the projection of the encoding $x \cdot G_u$ to the coordinates chosen by $\bar{S}$ uniquely identifies x. □

The lemma combined with a counting argument implies the following corollary:

Corollary 5.3. *Let $\mathcal{S}$ be any distribution on the subsets of $[n]$ of size at most m. Then all but a $\sqrt{5\epsilon}$ (resp., $\sqrt{5\epsilon'}$) fraction of the codes in $\mathcal{F}$ (resp., $\mathcal{G}$) can tolerate erasure patterns sampled from $\mathcal{S}$ with probability at least $1 - \sqrt{5\epsilon}$ (resp., $1 - \sqrt{5\epsilon'}$).* □

Note that the result holds irrespective of the distribution $\mathcal{S}$, contrary to the familiar case of $\mathsf{BEC}(p)$ for which the erasure pattern is an i.i.d. (i.e., independent and identically-distributed) sequence. For the case of $\mathsf{BEC}(p)$, the erasure pattern (regarded as its binary characteristic vector in $\mathbb{F}_2^n$) is given

by $S := (S_1, \ldots, S_n)$, where the random variables $S_1, \ldots, S_n \in \mathbb{F}_2$ are i.i.d. and $\Pr[S_i = 1] = p$. We denote this particular distribution by $\mathcal{B}_{n,p}$, which assigns a nonzero probability to every vector in $\mathbb{F}_2^n$. Thus in this case we cannot directly apply Corollary 5.3. However, note that $\mathcal{B}_{n,p}$ can be written as a convex combination

$$\mathcal{B}_{n,p} = (1-\gamma)\mathcal{U}_{n,\leq p'} + \gamma\mathcal{D}, \tag{5.1}$$

for $p' := p + \Omega(1)$ that is arbitrarily close to p, where $\mathcal{D}$ is an "error distribution" whose contribution γ is exponentially small. The distribution $\mathcal{U}_{n,\leq p'}$ is the distribution $\mathcal{B}_{n,p}$ conditioned on vectors of weight at most np'. Corollary 5.3 applies to $\mathcal{U}_{n,\leq p'}$ by setting $m = np'$. Moreover, by the convex combination above, the erasure decoding error probability of any code for erasure pattern distributions $\mathcal{B}_{n,p}$ and $\mathcal{U}_{n,\leq p'}$ differ by no more than γ. Therefore, the above result applied to the erasure distribution $\mathcal{U}_{n,\leq p'}$ handles the particular case of $\mathsf{BEC}(p)$ with essentially no change in the error probability.

In light of Corollary 5.3, in order to obtain rates arbitrarily close to the channel capacity, the output lengths of f and g must be sufficiently close to the entropy requirement m. More precisely, it suffices to have $r \leq (1+\alpha)m$ and $k \geq (1-\alpha)m$ for arbitrarily small constant $\alpha > 0$. The seed length of f and g determine the size of the code ensemble. Moreover, the error of the extractor and condenser determine the erasure error probability of the resulting code ensemble. As achieving the channel capacity is the most important concern for us, we will need to instantiate f (resp., g) with a linear, strong, lossless condenser (resp., extractor) whose output length is close to m. We mention one such instantiation for each function.

For both functions f and g, we can use the explicit extractor and lossless condenser obtained from the Leftover Hash Lemma (Lemma 2.17), which is optimal in the output length, but requires a large seed, namely, $d = n$. The ensemble resulting this way will thus have size 2^n, but attains a positive error exponent $\delta/2$ for an arbitrary rate loss $\delta > 0$. Using an optimal lossless condenser or extractor with seed length $d = \log(n) + O(\log(1/\epsilon))$ and output length close to m, it is possible to obtain a polynomially small capacity-achieving ensemble. However, in order to obtain an explicit ensemble of codes, the condenser of extractor being used must be explicit as well.

In the world of linear extractors, we can use Trevisan's extractor (Theorem 2.20) to improve the size of the ensemble compared to what obtained from the Leftover Hash Lemma. In particular, Trevisan's extractor combined with Corollary 5.3 (using ensemble $\mathcal{G}$) immediately gives the following result:

Corollary 5.4. *Let $p, c > 0$ be arbitrary constants. Then for every integer $n > 0$, there is an explicit ensemble $\mathcal{G}$ of linear codes of rate $1 - p - o(1)$ such that, the size of $\mathcal{G}$ is quasipolynomial, i.e., $|\mathcal{G}| = 2^{O(c^3 \log^3 n)}$, and, all but an $n^{-c} = o(1)$ fraction of the codes in the ensemble have error probability at most n^{-c} when used over $\mathsf{BEC}(p)$.* □

For the ensemble $\mathcal{F}$, on the other hand, we can use the linear lossless condenser of Guruswami et al. that only requires a logarithmic seed (Corollary 2.23). Using this condenser combined with Corollary 5.3, we can strengthen the above result as follows:

Corollary 5.5. *Let $p, c, \alpha > 0$ be arbitrary constants. Then for every integer $n > 0$, there is an explicit ensemble $\mathcal{F}$ of linear codes of rate $1 - p - \alpha$ such that $|\mathcal{G}| = O(n^{c'})$ for a constant c' only depending on c, α. Moreover, all but an $n^{-c} = o(1)$ fraction of the codes in the ensemble have error probability at most n^{-c} when used over $\mathsf{BEC}(p)$.* □

5.3 Codes for the Binary Symmetric Channel

The goal of this section is to design capacity achieving code ensembles for the binary symmetric channel $\mathsf{BSC}(p)$. In order to do so, we obtain codes for the general (and not necessarily memoryless) class $\mathsf{SC}(\mathbb{F}_q, \mathcal{Z})$ of symmetric channels, where $\mathcal{Z}$ is any flat distribution or sufficiently close to one. For concreteness, we will focus on the binary case where $q = 2$.

Recall that the capacity of $\mathsf{BSC}(\mathcal{Z})$, seen as a binary channel, is $1 - h(\mathcal{Z})$ where $h(\mathcal{Z})$ is the entropy rate of $\mathcal{Z}$. The special case $\mathsf{BSC}(p)$ is obtained by setting $\mathcal{Z} = \mathcal{B}_{n,p}$; i.e., the product distribution of n Bernoulli random variables with probability p of being equal to 1.

The code ensemble that we use for the symmetric channel is the ensemble $\mathcal{F}$, obtained from linear lossless condensers, that we introduced in the preceding section. Thus, we adopt the notation (and parameters) that we used before for defining the ensemble $\mathcal{F}$. Recall that each code in the ensemble has rate at least $1 - r/n$. In order to show that the ensemble is capacity achieving, we consider the following brute-force decoder for each code:

> *Brute-force decoder for code $\mathcal{C}_u$:* Given a received word $\hat{y} \in \mathbb{F}_2^n$, find a codeword $y \in \mathbb{F}_2^n$ of $\mathcal{C}_u$ used and a vector $z \in \mathsf{supp}(\mathcal{Z})$ such that $\hat{y} = y + z$. Output y, or an arbitrary codeword if no such pair is found. If there is more than one choice for the codeword y, arbitrarily choose one of them.

For each $u \in \mathbb{F}_2^d$, denote by $\mathcal{E}(\mathcal{C}_u, \mathcal{Z})$ the error probability of the above decoder for code $\mathcal{C}_u$ over $\mathsf{BSC}(\mathbb{F}_2, \mathcal{Z})$. The following lemma quantifies this probability:

Lemma 5.6. *Let $\mathcal{Z}$ be a flat distribution with entropy m. Then for at least a $1 - 2\sqrt{\epsilon}$ fraction of the choices of $u \in \mathbb{F}_2^d$, we have $\mathcal{E}(\mathcal{C}_u, \mathcal{Z}) \leq \sqrt{\epsilon}$.*

Proof. The proof is straightforward from the almost-injectivity property of lossless condensers discussed in Section 2.2.2. We will use this property to

construct a syndrome decoder for the code ensemble that achieves a sufficiently small error probability.

By Corollary 2.13, for a $1 - 2\sqrt{\epsilon}$ fraction of the choices of $u \in \{0,1\}^d$, the distribution $\mathcal{Y} := f(\mathcal{Z}, u)$ is $(\sqrt{\epsilon}/2)$-close to having min-entropy at least m. Fix any such u. We show that the error probability $\mathcal{E}(\mathcal{C}_u, \mathcal{Z})$ is bounded by $\sqrt{\epsilon}$.

For each $y \in \mathbb{F}_2^r$, define

$$\mathcal{N}(y) := |\{x \in \mathsf{supp}(\mathcal{Z})\colon f(x,u) = y\}|$$

and recall that $f(x,u) = H_u \cdot x$. Now suppose that a message is encoded using the code $\mathcal{C}_u$ to an encoding $x \in \mathcal{C}_u$, and that x is transmitted through the channel. The error probability $\mathcal{E}(\mathcal{C}_u, \mathcal{Z})$ can be written as

$$\begin{aligned}
\mathcal{E}(\mathcal{C}_u, \mathcal{Z}) &= \Pr_{z\sim\mathcal{Z}}[\exists x' \in \mathcal{C}_u, \exists z' \in \mathsf{supp}(\mathcal{Z}) \setminus z\colon x + z = x' + z'] \\
&\leq \Pr_{z\sim\mathcal{Z}}[\exists x' \in \mathcal{C}_u, \exists z' \in \mathsf{supp}(\mathcal{Z}) \setminus z\colon H_u \cdot (x+z) = H_u \cdot (x' + z')] \\
&= \Pr_{z\sim\mathcal{Z}}[\exists z' \in \mathsf{supp}(\mathcal{Z}) \setminus z\colon H_u \cdot z = H_u \cdot z'] \qquad (5.2) \\
&= \Pr_{z\sim\mathcal{Z}}[\mathcal{N}(H \cdot z) > 1] \\
&= \Pr_{z\sim\mathcal{Z}}[\mathcal{N}(f(x,u)) > 1], \qquad (5.3)
\end{aligned}$$

where (5.2) uses the fact that any codeword of $\mathcal{C}_u$ is in the right kernel of H_u.

By the first part of Proposition 2.14, there is a set $T \subseteq \mathbb{F}_2^r$ of size at least $(1 - \sqrt{\epsilon})|\mathsf{supp}(\mathcal{Z})|$ such that, $\mathcal{N}(y) = 1$ for every $y \in T$. Since $\mathcal{Z}$ is uniformly distributed on its support, this combined with (5.3) immediately implies that $\mathcal{E}(\mathcal{C}_u, \mathcal{Z}) \leq \sqrt{\epsilon}$. □

The lemma implies that any linear lossless condenser with entropy requirement m can be used to construct an ensemble of codes such that all but a small fraction of the codes are good for reliable transmission over $\mathsf{BSC}(\mathcal{Z})$, where $\mathcal{Z}$ is an arbitrary flat distribution with entropy at most m. Similar to the case of BEC, the seed length determines the size of the ensemble, the error of the condenser bounds the error probability of the decoder, and the output length determines the proximity of the rate to the capacity of the channel. Again, using the condenser given by the Leftover Hash Lemma (Lemma 2.17), we can obtain a capacity achieving ensemble of size 2^n. Moreover, using the linear lossless condenser of Guruswami et al. (Corollary 2.23) the ensemble can be made polynomially small (similar to the result given by Corollary 5.5).

It is not hard to see that the converse of the above result is also true; namely, that any ensemble of linear codes that is universally capacity achieving with respect to any choice of the noise distribution $\mathcal{Z}$ defines a strong linear, lossless, condenser. This is spelled out in the lemma below.

Lemma 5.7. *Let $\{\mathcal{C}_1, \ldots, \mathcal{C}_T\}$ be a binary code ensemble of length n and dimension $n - r$ such that for every flat distribution $\mathcal{Z}$ with min-entropy at most m on $\mathbb{F}_2^n$, all but a γ fraction of the codes in the ensemble (for some $\gamma \in [0, 1)$) achieve error probability at most ϵ (under syndrome decoding) when used over $\mathsf{SC}(\mathbb{F}_{q^n}, \mathcal{Z})$. Then the function $f \colon \mathbb{F}_2^n \times [T] \to \mathbb{F}_2^r$ defined as*

$$f(x, u) := H_u \cdot x,$$

where H_u is a parity check matrix of $\mathcal{C}_u$, is a strong, lossless, $(m, 2\epsilon + \gamma)$-condenser.

Proof. The proof is straightforward using similar arguments as in Lemma 5.6. Without loss of generality (by Proposition 2.8), let $\mathcal{Z}$ be a flat distribution with min-entropy m, and denote by $D \colon \mathbb{F}_2^r \to \mathbb{F}_2^n$ the corresponding syndrome decoder. Moreover, without loss of generality we have taken the decoder to be a deterministic function. For a randomized decoder, one can fix the internal coin flips so as to preserve the upper bound on its error probability. Now let u be chosen such that $\mathcal{C}_u$ achieves an error probability at most ϵ (we know this is the case for at least γT of the choices of u).

Denote by $T \subseteq \mathsf{supp}(\mathcal{Z})$ the set of noise realizations that can potentially confuse the syndrome decoder. Namely,

$$T := \{z \in \mathsf{supp}(\mathcal{Z}) \colon \exists z' \in \mathsf{supp}(\mathcal{Z}), z' \neq z, H_u \cdot z = H_u \cdot z'\}.$$

Note that, for a random $Z \sim \mathcal{Z}$, conditioned on the event that $Z \in T$, the probability that the syndrome decoder errs on Z is at least $1/2$, since we know that Z can be confused by at least one different noise realization. We can write this more precisely as

$$\Pr_{Z \sim \mathcal{Z}}[D(Z) \neq Z \mid Z \in T] \geq 1/2.$$

Since the error probability of the decoder is upper bounded by ϵ, we conclude that

$$\Pr_{Z \sim \mathcal{Z}}[Z \in T] \leq 2\epsilon.$$

Therefore, the fraction of the elements on support of $\mathcal{Z}$ that collide with some other element under the mapping defined by H_u is at most 2ϵ. Namely,

$$|\{H_u \cdot z \colon z \in \mathsf{supp}(\mathcal{Z})\}| \geq 2^m(1 - 2\epsilon),$$

and this is true for at least $1 - \gamma$ fraction of the choices of u. Thus, for a uniformly random $U \in [T]$ and $Z \sim \mathcal{Z}$, the distribution of $(U, H_U \cdot Z)$ has a support of size at least

$$(1 - \gamma)(1 - 2\epsilon)T2^m \geq (1 - \gamma - 2\epsilon)T2^m.$$

By the second part of Proposition 2.14, we conclude that this distribution is $(2\epsilon + \gamma)$-close to having entropy $m + \log T$ and thus, the function f defined in the statement is a strong lossless $(m, 2\epsilon + \gamma)$-condenser. □

By this lemma, the known lower bounds on the seed length and the output length of lossless condensers that we discussed in Chapter 2 translate into lower bounds on the size of the code ensemble and proximity to the capacity that can be obtained from our framework. In particular, in order to get a positive error exponent (i.e., exponentially small error in the block length), the size of the ensemble must be exponentially large.

It is worthwhile to point out that the code ensembles $\mathcal{F}$ and $\mathcal{G}$ discussed in this and the preceding section preserve their erasure and error correcting properties under any change of basis in the ambient space $\mathbb{F}_2^n$, due to the fact that a change of basis applied on any linear condenser results in a linear condenser with the same parameters. This is a property achieved by the trivial, but large, ensemble of codes defined by the set of all $r \times n$ parity check matrices. Observe that no single code can be universal in this sense, and it is inevitable to have a sufficiently large ensemble to attain this property.

The Case $\mathsf{BSC}(p)$

For the special case of $\mathsf{BSC}(p)$, the noise distribution $\mathcal{B}_{n,p}$ is not a flat distribution. Fortunately, similar to the BEC case, we can again use convex combinations to show that the result obtained in Lemma 5.6 can be extended to this important noise distribution. The main tool that we need is an extension of Lemma 5.6 to convex combinations with a small number of components.

Suppose that the noise distribution $\mathcal{Z}$ is not a flat distribution but can be written as a convex combination

$$\mathcal{Z} = \alpha_1 \mathcal{Z}_1 + \cdots + \alpha_t \mathcal{Z}_t. \tag{5.4}$$

of t flat distributions, where the number t of summands is not too large, and

$$|\mathsf{supp}(\mathcal{Z}_1)| \geq |\mathsf{supp}(\mathcal{Z}_2)| \geq \cdots \geq |\mathsf{supp}(\mathcal{Z}_t)|.$$

For this more general case, we need to slightly tune our brute-force decoder in the way it handles ties. In particular, we now require the decoder to find a codeword $y \in \mathcal{C}_u$ and a potential noise vector $z \in \mathsf{supp}(\mathcal{Z})$ that add up to the received word, as before. However, in case more than one matching pair is found, we will require the decoder to choose the one whose noise vector z belongs to the component $\mathcal{Z}_1, \ldots, \mathcal{Z}_t$ with smallest support (i.e., largest index). If the noise vector $z \in \mathsf{supp}(\mathcal{Z}_i)$ that maximizes the index i is still not unique, the decoder can arbitrarily choose one. Under these conventions, we can now prove the following:

Lemma 5.8. *Suppose that a noise distribution $\mathcal{Z}$ is as in* (5.4)*, where each component $\mathcal{Z}_i$ has entropy at most m, and the function f defining the ensemble $\mathcal{F}$ is a strong lossless $(\leq m+1, \epsilon)$-condenser. Then for at least a $1 - t(t+1)\sqrt{\epsilon}$ fraction of the choices of $u \in \mathbb{F}_2^d$, the brute-force decoder satisfies $\mathcal{E}(\mathcal{C}_u, \mathcal{Z}) \leq 2t\sqrt{\epsilon}$.*

Proof. For each $1 \leq i \leq j \leq t$, we define a flat distribution $\mathcal{Z}_{ij}$ that is uniformly supported on $\mathsf{supp}(\mathcal{Z}_i) \cup \mathsf{supp}(\mathcal{Z}_j)$. Observe that each $\mathcal{Z}_{ij}$ has min-entropy at most $m+1$ and thus the function f is a lossless condenser with error at most ϵ for this source. By Corollary 2.13 and a union bound, for a $1 - t(t+1)\sqrt{\epsilon}$ fraction of the choices of $u \in \{0,1\}^d$, all $t(t+1)/2$ distributions

$$f(\mathcal{Z}_{ij}, u) \colon 1 \leq i \leq j \leq t$$

are simultaneously $(\sqrt{\epsilon}/2)$-close to having min-entropy at least m. Fix any such u.

Consider a random variable Z, representing the channel noise, that is sampled from $\mathcal{Z}$ as follows: First choose an index $I \in [t]$ randomly according to the distribution induced by $(\alpha_1, \ldots, \alpha_t)$ over the indices, and then sample a random noise $Z \sim \mathcal{Z}_I$. Using the same line of reasoning leading to (5.2) in the proof of Lemma 5.6, the error probability with respect to the code $\mathcal{C}_u$ (i.e., the probability that the tuned distance decoder gives a wrong estimate on the noise realization Z) can now be bounded as

$$\mathcal{E}(\mathcal{C}_u, \mathcal{Z}) \leq \Pr_{I,Z}[\exists i \in \{I, \ldots, t\}, \exists z' \in \mathsf{supp}(\mathcal{Z}_i) \setminus Z \colon f(Z,u) = f(z',u)].$$

For $i = 1, \ldots, t$, denote by $\mathcal{E}_i$ the right hand side probability in the above bound conditioned on the event that $I = i$. Fix any choice of the index i. Now it suffices to obtain an upper bound on $\mathcal{E}_i$ irrespective of the choice of i, since

$$\mathcal{E}(\mathcal{C}_u, \mathcal{Z}) \leq \sum_{i \in [t]} \alpha_i \mathcal{E}_i.$$

We call a noise realization $z \in \mathsf{supp}(\mathcal{Z}_i)$ *confusable* if

$$\exists j \geq i, \exists z' \in \mathsf{supp}(\mathcal{Z}_j) \setminus z \colon f(z,u) = f(z',u).$$

That is, a noise realization is confusable if it can potentially cause the brute-force decoder to compute a wrong noise estimate. Our goal is to obtain an upper bound on the fraction of vectors on $\mathsf{supp}(\mathcal{Z}_i)$ that are confusable.

For each $j \geq i$, we know that $f(\mathcal{Z}_{ij}, u)$ is $(\sqrt{\epsilon}/2)$-close to having min-entropy at least m. Therefore, by the first part of Proposition 2.14, the set of confusable elements

$$\{z \in \mathsf{supp}(\mathcal{Z}_i) \colon \exists z' \in \mathsf{supp}(\mathcal{Z}_j) \setminus z \text{ such that } f(z,u) = f(z',u)\}$$

has size at most $\sqrt{\epsilon}|\mathsf{supp}(\mathcal{Z}_{ij})| \leq 2\sqrt{\epsilon}|\mathsf{supp}(\mathcal{Z}_i)|$ (using the fact that, since $j \geq i$, the support of $\mathcal{Z}_j$ is no larger than that of $\mathcal{Z}_i$). By a union bound on the choices of j, we see that the fraction of confusable elements on $\mathsf{supp}(\mathcal{Z}_i)$ is at most $2t\sqrt{\epsilon}$. Therefore, $\mathcal{E}_i \leq 2t\sqrt{\epsilon}$ and we get the desired upper bound on the error probability of the brute-force decoder. $\square$

The result obtained by Lemma 5.8 can be applied to the channel $\mathsf{BSC}(p)$ by observing that the noise distribution $\mathcal{B}_{n,p}$ can be written as a convex combination

$$\mathcal{B}_{n,p} = \sum_{i=n(p-\eta)}^{n(p+\eta)} \alpha_i \mathcal{U}_{n,i} + \gamma \mathcal{D},$$

where $\mathcal{U}_{n,i}$ denotes the flat distribution supported on binary vectors of length n and Hamming weight exactly i, and $\mathcal{D}$ is the distribution $\mathcal{B}_{n,p}$ conditioned on the vectors whose Hamming weights lie outside the range $[n(p-\eta), n(p+\eta)]$. The parameter $\eta > 0$ can be chosen as an arbitrarily small real number, so that the min-entropies of the distributions $\mathcal{U}_{n,i}$ become arbitrarily close to the Shannon entropy of $\mathcal{B}_{n,p}$; namely, $nh(p)$. This can be seen by the estimate

$$\binom{n}{w} = 2^{nh(w/n) \pm o(n)},$$

$h(\cdot)$ being the binary entropy function, that is easily derived from Stirling's formula. By Chernoff bounds, the error γ can be upper bounded as

$$\gamma = \Pr_{Z \sim \mathcal{B}_{n,p}} [|\mathsf{wgt}(Z) - np| > \eta n] \leq 2e^{-c_\eta np} = 2^{-\Omega(n)},$$

where $c_\eta > 0$ is a constant only depending on η, and is thus exponentially small. Thus the error probability attained by any code under noise distributions $\mathcal{B}_{n,p}$ and $\mathcal{Z} := \sum_{i=n(p-\eta)}^{n(p+\eta)} \alpha_i \mathcal{U}_{n,i}$ differ by the exponentially small quantity γ. We may now apply Lemma 5.8 on the noise distribution $\mathcal{Z}$ to attain code ensembles for the binary symmetric channel $\mathsf{BSC}(p)$. The error probability of the ensemble is at most $2n\sqrt{\epsilon}$, and this bound is satisfied by at least a $1 - n^2\sqrt{\epsilon}$ fraction of the codes. Finally, the code ensemble is capacity achieving for $\mathsf{BSC}(p)$ provided that the condenser f attains an output length $r \leq (1+\alpha)(p+\eta)n$ for arbitrarily small constant α, and $\epsilon = o(n^{-4})$.

5.4 Explicit Capacity Achieving Codes

In the preceding sections, we showed how to obtain small ensembles of explicit capacity achieving codes for various discrete channels, including the important special cases $\mathsf{BEC}(p)$ and $\mathsf{BSC}(p)$. Two drawbacks related to these constructions are:

1. While an overwhelming fraction of the codes in the ensemble are capacity achieving, in general it is not clear how to pin down a single, capacity achieving code in the ensemble.

2. For the symmetric additive noise channels, the brute-force decoder is extremely inefficient and is of interest only for proving that the constructed ensembles are capacity achieving.

In a classic work, Justesen [87] showed that the idea of code concatenation[5] first introduced by Forney [60] can be used to transform any ensemble of capacity achieving codes, for a memoryless channel, into an explicit, efficiently decodable code with improved error probability over the same channel. In this section we revisit this idea and apply it to our ensembles. For concreteness, we focus on the binary case and consider a memoryless channel $\mathcal{C}$ that is either $\mathsf{BEC}(p)$ or $\mathsf{BSC}(p)$.

Throughout this section, we consider an ensemble $\mathcal{S}$ of linear codes with block length n and rate R, for which it is guaranteed that all but a $\gamma = o(1)$ fraction of the codes are capacity achieving (for a particular DMSC, in our case either $\mathsf{BEC}(p)$ or $\mathsf{BSC}(p)$) with some vanishing error probability $\eta = o(1)$ (the asymptotics are considered with respect to the block length n).

Justesen's concatenated codes take an outer code $\mathcal{C}_{\text{out}}$ of block length $s := |\mathcal{S}|$, alphabet $\mathbb{F}_{2^k}$, rate R' as the *outer code*. The particular choice of the outer code in the original construction is Reed-Solomon codes. However, we point out that any outer code that allows unique decoding of some constant fraction of errors at rates arbitrarily close to one would suffice for the purpose of constructing capacity achieving codes. In particular, in this section we will use an expander-based construction of asymptotically good codes due to Spielman [141], from which the following theorem can be easily derived[6]:

Theorem 5.9. *For every integer $k > 0$ and every absolute constant $R' < 1$, there is an explicit family of $\mathbb{F}_2$-linear codes over $\mathbb{F}_{2^k}$ for every block length and rate R' that is error-correcting for an $\Omega(1)$ fraction of errors. The running time of the encoder and the decoder is linear in the bit-length of the codewords.*

5.4.1 Justesen's Concatenation Scheme

The concatenation scheme of Justesen differs from traditional concatenation in that the outer code is concatenated with an ensemble of codes rather than a single inner code.

In this construction, size of the ensemble is taken to be matching with the block length of the outer code, and each symbol of the outer code is encoded with one of the inner codes in the ensemble. We use the notation $\mathcal{C} := \mathcal{C}_{\text{out}} \diamond \mathcal{S}$ to denote concatenation of an outer code $\mathcal{C}_{\text{out}}$ with the ensemble $\mathcal{S}$ of inner codes. Suppose that the alphabet size of the outer code is taken as $2^{\lfloor Rn \rfloor}$, where we recall that n and R denote the block length and rate of the inner codes in $\mathcal{S}$.

The encoding of a message with the concatenated code can be obtained as follows: First, the message is encoded using $\mathcal{C}_{\text{out}}$ to obtain an encoding $(c_1, \ldots, c_s) \in \mathbb{F}_{2^k}^s$, where $k = \lfloor Rn \rfloor$ denotes the dimension of the inner codes.

[5] A quick review of code concatenation and its basic properties appears in Appendix A.

[6] There are alternative choices of the outer code that lead to a similar result, e.g., expander-based codes due to Guruswami and Indyk [76].

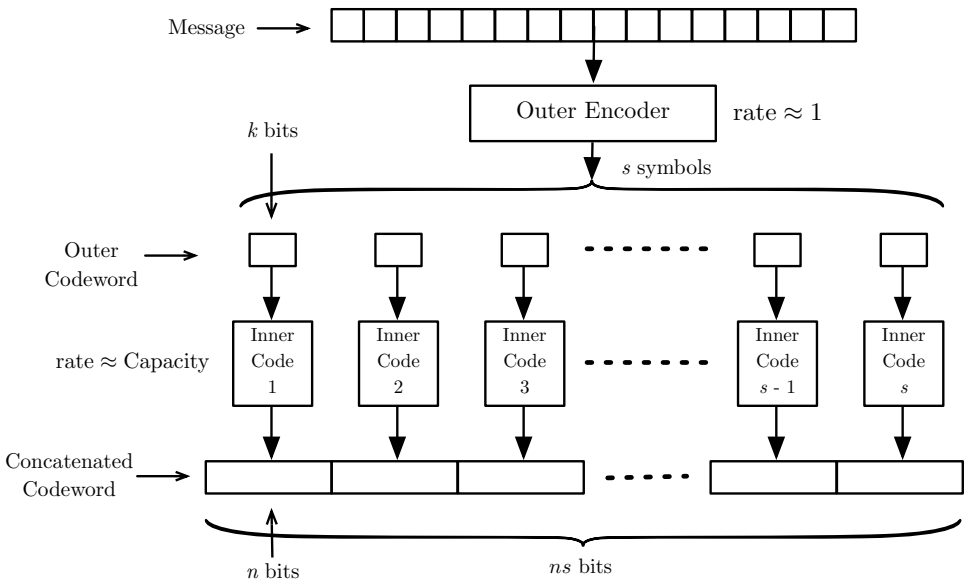


Figure 5.3: Justesen's concatenation scheme.

Then, for each $i \in [s]$, the ith symbol of the encoding c_i is further encoded by the ith code in the ensemble $\mathcal{S}$ (under some arbitrary ordering of the codes in the ensemble), resulting in a binary sequence c'_i of length n. The ns-bit long binary sequence $(c'_1, \ldots, c'_s)$ defines the encoding of the message under $\mathcal{C}_{\text{out}} \diamond \mathcal{S}$. The concatenation is scheme is depicted in Figure 5.3.

Similar to classical concatenated codes, the resulting binary code $\mathcal{C}$ has block length $N := ns$ and dimension $K := kk'$, where k' is the dimension of the outer code $\mathcal{C}_{\text{out}}$. However, the neat idea in Justesen's concatenation is that it eliminates the need for a brute-force search for finding a good inner code, as long as almost all inner codes are guaranteed to be good.

5.4.2 The Analysis

In order to analyze the error probability attained by the concatenated code $\mathcal{C}_{\text{out}} \diamond \mathcal{S}$, we consider the following naive decoder[7]:

1. Given a received sequence $(y_1, \ldots, y_s) \in (\mathbb{F}_2^n)^s$, apply an appropriate decoder for the inner codes (e.g., the brute-force decoder for BSC, or Gaussian elimination for BEC) to decode each y_i to a codeword c'_i of the ith code in the ensemble.

[7] Alternatively, one could use methods such as Forney's Generalized Minimum Distance (GMD) decoder for Reed-Solomon codes [60]. However, the naive decoder suffices for our purposes and works for any asymptotically good choice of the outer code.

2. Apply the outer code decoder on $(c'_1, \ldots, c'_s)$ that is guaranteed to correct some constant fraction of errors, to obtain a codeword $(c_1, \ldots, c_s)$ of the outer code $\mathcal{C}_{\text{out}}$.

3. Recover the decoded sequence from the corrected encoding $(c_1, \ldots, c_s)$.

Since the channel is assumed to be memoryless, the noise distributions on inner codes are independent. Let $\mathcal{G} \subseteq [s]$ denote the set of coordinate positions corresponding to "good" inner codes in $\mathcal{S}$ that achieve an error probability bounded by η. By assumption, we have $\mathcal{G} \geq (1-\gamma)|\mathcal{S}|$.

Suppose that the outer code $\mathcal{C}_{\text{out}}$ corrects some $\gamma + \alpha$ fraction of adversarial errors, for a constant $\alpha > \eta$. Then an error might occur only if more than αN of the codes in $\mathcal{G}$ fail to obtain a correct decoding. We expect the number of failures within the good inner codes to be $\eta|\mathcal{G}|$. Due to the noise independence, it is possible to show that the fraction of failures may deviate from the expectation η only with a negligible probability. In particular, a direct application of Chernoff bound implies that the probability that more than an α fraction of the good inner codes err is at most

$$\eta^{\alpha'|\mathcal{G}|} = 2^{-\Omega_\alpha(\log(1/\eta)s)}, \tag{5.5}$$

where $\alpha' > 0$ is a constant that only depends on α. This also upper bounds the error probability of the concatenated code. In particular, we see that if the error probability η of the inner codes is exponentially small in their block length n, the concatenated code also achieves an exponentially small error in its block length N.

Now we analyze the encoding and decoding complexity of the concatenated code, assuming that Spielman's expander codes (Theorem 5.9) are used for the outer code. With this choice, the outer code becomes equipped with a linear-time encoder and decoder. Since any linear code can be encoded in quadratic time (in its block length), the concatenated code can be encoded in $O(n^2 s)$, which for $s \gg n$ can be considered "almost linear" in the block length $N = ns$ of $\mathcal{C}$. The decoding time of each inner code is cubic in n for the erasure channel, since decoding reduces to Gaussian elimination, and thus for this case the naive decoder runs in time $O(n^3 s)$. For the symmetric channel, however, the brute-force decoder used for the inner codes takes exponential time in the block length, namely, $2^{Rn}\mathsf{poly}(n)$. Therefore, the running time of the decoder for concatenated code becomes bounded by $O(2^{Rn} s\mathsf{poly}(n))$. When the inner ensemble is exponentially large; i.e., $s = 2^n$ (which is the case for our ensembles if we use the Leftover Hash Lemma), the decoding complexity becomes $O(s^{1+R}\mathsf{poly}(\log s))$ which is at most quadratic in the block length of $\mathcal{C}$.

Since the rate R' of the outer code can be made arbitrarily close to 1, rate of the concatenated code $\mathcal{C}$ can be made arbitrarily close to the rate R of the inner codes. Thus, if the ensemble of inner codes is capacity-achieving, so would be the concatenated code.

5.4.3 Density of the Explicit Family

In the preceding section we saw how to obtain explicit capacity achieving codes from capacity achieving code ensembles using concatenation. One of the important properties of the resulting family of codes that is influenced by the size of the inner code ensemble is the set of block lengths N for which the concatenated code is defined. Recall that $N = ns$, where n and s respectively denote the block length of the inner codes and the size of the code ensemble, and the parameter s is a function of n. For instance, for all classical examples of capacity achieving code ensembles (namely, Wozencraft's ensemble, Goppa codes and shortened cyclic codes) we have $s(n) = 2^n$. In this case, the resulting explicit family of codes would be defined for integer lengths of the form $N(i) = i2^i$.

A trivial approach for obtaining capacity achieving codes for all lengths is to use a *padding trick*. Suppose that we wish to transmit a particular bit sequence of length K through the channel using the concatenated code family of rate ρ that is taken to be sufficiently close to the channel capacity. The sequence might originate from a source that does not produce a constant stream of bits (e.g., consider a terminal emulator that produces data only when a user input is available).

Ideally, one requires the length of the encoded sequence to be $N = \lceil K/\rho \rceil$. However, since the family might not be defined for the block length N, we might be forced to take a code $\mathcal{C}$ in the family with smallest length $N' \geq N$ that is of the form $N' = ns(n)$, for some integer n, and pad the original message with redundant symbols. This way we have encoded a sequence of length K to one of length N', implying an effective rate K/N'. The rate loss incurred by padding is thus equal to $\rho - K/N' = K(1/N - 1/N')$. Thus, if $N' \geq N(1+\delta)$ for some positive constant $\delta > 0$, the rate loss becomes lower bounded by a constant and thus, even if the original concatenated family is capacity achieving, it no longer remains capacity achieving when extended to arbitrarily chosen lengths using the padding trick.

Therefore, if we require the explicit family obtained from concatenation to remain capacity achieving for all lengths, the set of block lengths $\{is(i)\}_{i \in \mathbb{N}}$ for which it is defined must be sufficiently dense. This is the case provided that we have

$$\frac{s(n)}{s(n+1)} = 1 - o(1),$$

which in turn, requires the capacity achieving code ensemble to have a sub-exponential size (by which we mean $s(n) = 2^{o(n)}$).

Using the framework introduced in this chapter, linear extractors and lossless condensers that achieve nearly optimal parameters would result in code ensembles of polynomial size in n. The explicit erasure code ensemble obtained from Trevisan's extractor (Corollary 5.4) or Guruswami-Umans-Vadhan's lossless condenser (Corollary 5.5) combined with Justesen's concatenation scheme

results in an explicit sequence of capacity achieving codes for the binary erasure channel that is defined for every block length, and allows almost linear-time (i.e., $N^{1+o(1)}$) encoding and decoding. Moreover, the latter sequence of codes that is obtained from a lossless condenser is capacity achieving for the binary symmetric channel (with a matching bit-flip probability) as well.

5.5 Duality of Linear Affine Condensers

In Section 5.2 we saw that linear extractors for bit-fixing sources can be used to define generator matrices of a family of erasure-decodable codes. On the other hand, we showed that linear lossless condensers for bit-fixing sources define parity check matrices of erasure-decodable codes.

Recall that generator and parity check matrices are dual notions, and in our construction we have considered matrices in one-to-one correspondence with linear mappings. Indeed, we have used linear mappings defined by extractors and lossless condensers to obtain generator and parity check matrices of our codes (where the ith row of the matrix defines the coefficient vector of the linear form corresponding to the ith output of the mapping). Thus, we get a natural duality between linear functions: If two linear functions represent generator and parity check matrices of the same code, they can be considered dual[8]. Just in the same way that the number of rows of a generator matrix and the corresponding parity check matrix add up to their number of columns (provided that there is no linear dependence between the rows), the dual of a linear function mapping $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$ (where $m \leq n$) that has no linear dependencies among its $n-m$ outputs can be taken to be a linear function mapping $\mathbb{F}_q^n$ to $\mathbb{F}_q^{n-m}$.

In fact, a duality between linear extractors and lossless condensers for affine sources is implicit in the analysis leading to Corollary 5.3. Namely, it turns out that if a linear function is an extractor for an affine source, the dual function becomes a *lossless condenser* for the *dual distribution*, and vice versa. This is made precise (and slightly more general) in the following theorem.

Theorem 5.10. *Suppose that the linear mapping defined by a matrix $G \in \mathbb{F}_q^{m\times n}$ of rank $m \leq n$ is a $(k \log q) \to_0 (k' \log q)$ condenser for a k-dimensional affine source $\mathcal{X}$ over $\mathbb{F}_q^n$ so that for $X \sim \mathcal{X}$, the distribution of $G \cdot X^\top$ has entropy at least $k' \log q$. Let $H \in \mathbb{F}_q^{(n-m)\times n}$ be a dual matrix for G (i.e., $GH^\top = 0$) of rank $n-m$ and $\mathcal{Y}$ be an $(n-k)$-dimensional affine space over $\mathbb{F}_q^n$ supported on a translation of the dual subspace corresponding to the support of $\mathcal{X}$. Then for $Y \sim \mathcal{Y}$, the distribution of $H \cdot Y^\top$ has entropy at least $(n-k+k'-m)\log q$.*

[8]Note that, under this notion of duality, the dual of a linear function need not be unique even though its linear-algebraic properties (e.g., kernel) would be independent of its choice.

Proof. Suppose that $\mathcal{X}$ is supported on a set

$$\{x \cdot A_G + a \colon x \in \mathbb{F}_q^k\},$$

where $A_G \in \mathbb{F}_q^{k \times n}$ has rank k and $a \in \mathbb{F}_q^n$ is a fixed row vector. Moreover we denote the dual distribution $\mathcal{Y}$ by the set

$$\{y \cdot A_H + b \colon y \in \mathbb{F}_q^{n-k}\},$$

where $b \in \mathbb{F}_q^n$ is fixed and $A_H \in \mathbb{F}_q^{(n-k) \times n}$ is of rank $n-k$, and we have the orthogonality relationship $A_H \cdot A_G^\top = 0$.

The assumption that G is a $(k \log q) \to_0 (k' \log q)$-condenser implies that the distribution

$$G \cdot (A_G^\top \cdot \mathcal{U}_{\mathbb{F}_q^k} + a^\top),$$

where $\mathcal{U}_{\mathbb{F}_q^k}$ stands for a uniformly random row vector in $\mathbb{F}_q^k$, is an affine source of dimension at least k', equivalent to saying that the matrix $G \cdot A_G^\top \in \mathbb{F}_q^{m \times k}$ has rank at least k' (since rank is equal to the dimension of the image), or in symbols,

$$\mathsf{rank}(G \cdot A_G^\top) \geq k'. \tag{5.6}$$

Observe that since we have assumed $\mathsf{rank}(G) = m$, its right kernel is $(n-m)$-dimensional, and thus the linear mapping defined by G cannot reduce more than $n-m$ dimensions of the affine source $\mathcal{X}$. Thus, the quantity $n-k+k'-m$ is non-negative.

By a similar argument as above, in order to show the claim we need to show that

$$\mathsf{rank}(H \cdot A_H^\top) \geq n - k + k' - m.$$

Suppose not. Then the right kernel of $H \cdot A_H^\top \in \mathbb{F}_q^{(n-m) \times (n-k)}$ must have dimension larger than $(n-k) - (n-k+k'-m) = m - k'$. Denote this right kernel by $\mathcal{R} \subseteq \mathbb{F}_q^{n-k}$. Since the matrix A_H is assumed to have maximal rank $n-k$, and $n-k \geq m-k'$, for each nonzero $y \in \mathcal{R}$, the vector $y \cdot A_H \in \mathbb{F}_q^n$ is nonzero and since $H \cdot (A_H^\top y^\top) = 0$ (by the definition of right kernel), the duality of G and H implies that there is a nonzero $x \in \mathbb{F}_q^m$ where

$$x \cdot G = y \cdot A_H,$$

and the choice of y uniquely specifies x. In other words, there is a subspace $\mathcal{R}' \subseteq \mathbb{F}_q^m$ such that

$$\dim(\mathcal{R}') = \dim(\mathcal{R}),$$

and

$$\{x \cdot G \colon x \in \mathcal{R}'\} = \{y \cdot A_H \colon y \in \mathcal{R}\}.$$

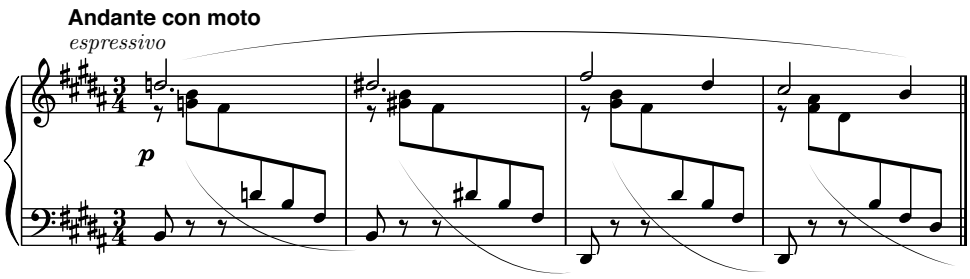
But observe that, by orthogonality of A_G and A_H, every y satisfies $y \cdot A_H A_G^\top = 0$, meaning that for every $x \in \mathcal{R}'$, we must have $x \cdot GA_G^\top = 0$. Thus the left kernel of $GA_G^\top$ has dimension larger than $m - k'$ (since $\mathcal{R}'$ does), and we conclude that the matrix $GA_G^\top$ has rank less than k', a contradiction for (5.6). □

Since every k-dimensional affine space over $\mathbb{F}_q^n$ has an $(n-k)$-dimensional dual vector space, the above result combined with Proposition 5.1 directly implies the following corollary:

Corollary 5.11. *Suppose that the linear mapping defined by a matrix $G \in \mathbb{F}_q^{m \times n}$ of rank $m \leq n$ is a $(k \log q) \to_\epsilon (k' \log q)$ condenser, for some $\epsilon < 1/2$. Let $H \in \mathbb{F}_q^{(n-m) \times n}$ of rank $n - m$ be so that $GH^\top = 0$. Then, the linear mapping defined by H is an $(n-k) \log q \to_0 (n-k+k'-m) \log q$ condenser.* □

Similarly, linear *seeded* condensers for affine sources define linear seeded *dual condensers* for affine sources with complementary entropy (this is done by taking the dual linear function for every fixing of the seed).

Two important special cases of the above results are related to affine extractors and lossless condensers. When the linear mapping G is an affine extractor for k-dimensional distributions, the dual mapping H becomes a lossless condenser for $(n-k)$-dimensional spaces, and vice versa.



Johannes Brahms (1833–1897): Ballade Op. 10 No. 4 in B major.

"I confess that Fermat's Theorem as an isolated proposition has very little interest for me, because I could easily lay down a multitude of such propositions, which one could neither prove nor dispose of."

— *Carl Friedrich Gauss*

Chapter 6

Codes on the Gilbert-Varshamov Bound

One of the central problems in coding theory is the construction of codes with extremal parameters. Typically, one fixes an alphabet size q, and two among the three fundamental parameters of the code (block-length, number of codewords, and minimum distance), and asks about extremal values of the remaining parameter such that there is a code over the given alphabet with the given parameters. For example, fixing the minimum distance d and the block-length n, one may ask for the largest number of codewords M such that there exists a code over an alphabet with q elements having n, M, d as its parameters, or in short, an $(n, M, d)_q$-code.

Answering this question in its full generality is extremely difficult, especially when the parameters are large. For this reason, researchers have concentrated on asymptotic assertions: to any $[n, \log M, d]_q$-code C we associate a point $(\delta(C), R(C)) \in [0,1]^2$, where $\delta(C) = d/n$ and $R(C) = \log_q M/n$ are respecitvely the relative distance and rate of the code. A particular point (δ, R) is called *asymptotically achievable* (over a q-ary alphabet) if there exists a sequence $(C_1, C_2, \ldots)$ of codes of increasing block-length such that $\delta(C_i) \to \delta$ and $R(C_i) \to R$ as $i \to \infty$.

Even with this asymptotic relaxation the problem of determining the shape of the set of asymptotically achievable points remains difficult. Let $\alpha_q(\delta)$ be defined as the supremum of all R such that (δ, R) is asymptotically achievable over a q-ary alphabet. It is known that α_q is a continuous function of δ [105], that $\alpha_q(0) = 1$ (trivial), and $\alpha_q(\delta) = 0$ for $\delta \geq (q-1)/q$ (by the Plotkin bound). However, for no $\delta \in (0, (q-1)/q)$ and for no q is the value of $\alpha_q(\delta)$ known.

What is known are lower and upper bounds for α_q. The best lower bound known is due to Gilbert and Varshamov[68, 157] which states that $\alpha_q(\delta) \geq$

$1 - h_q(\delta)$, where the q-ary entropy function h_q is defined as

$$h_q(\delta) := -\delta \log_q \delta - (1-\delta)\log_q(1-\delta) + \delta \log_q(q-1).$$

Up until 1982, years of research had made it plausible to think that this bound is tight, i.e., that $\alpha_q(\delta) = 1 - h_q(\delta)$. Goppa's invention of algebraic-geometric codes [72], and the subsequent construction of Tsfasman, Vlăduţ, and Zink [154] using curves with many points over a finite field and small genus showed however that the bound is not tight when the alphabet size is large enough. Moreover, Tsfasman et al. also gave a polynomial time construction of such codes (which has been greatly simplified since, see, e.g., [67]).

The fate of the binary alphabet is still open. Many researchers still believe that $\alpha_2(\delta) = 1 - h_2(\delta)$. In fact, for a randomly chosen linear code C (one in which the entries of a generator matrix are chosen independently and uniformly over the alphabet) and for any positive ϵ we have $R(C) \geq 1 - h_q(\delta(C)) - \epsilon$ with high probability (with probability at least $1 - 2^{-nc_\epsilon}$ where n is the block-length and c_ϵ is a constant depending on ϵ). However, even though this shows that most randomly chosen codes are arbitrarily close to the Gilbert-Varshamov bound, no explicit polynomial time construction of such codes is known when the alphabet size is small (e.g., for binary alphabets).

In this chapter, we use the technology of pseudorandom generators which has played a prominent role in the theoretical computer science research in recent years to (conditionally) produce, for any block-length n and any rate $R < 1$, a list of $\mathsf{poly}(n)$ many codes of block length n and designed rate R (over an arbitrary alphabet) such that a very large fraction of these codes has parameters arbitrarily close to the Gilbert-Varshamov bound. Here, $\mathsf{poly}(n)$ denotes a polynomial in n.

In a nutshell, our construction is based on the pseudorandom generator of Nisan and Wigderson [115]. In particular, we will first identify a Boolean function f of which we assume that it satisfies a certain complexity-theoretic assumption. More precisely, we assume that the function cannot be computed by algorithms that require sub-exponential amount of memory. A natural candidate for such a function is given later in the chapter. This function is then extended to produce nk bits from $O(\log n)$ bits. The extended function is called a *pseudorandom generator*. The main point about this extended function is that the nk bits produced cannot be distinguished from random bits by a Turing machine with restricted resources. In our case, the output cannot be distinguished from a random sequence when a Turing machine is used which uses only an amount of space that is polynomially bounded in the length of its input.

The new nk bits are regarded as the entries of a generator matrix of a code. Varying the base $O(\log n)$ bits in all possible ways gives us a polynomially long

list of codes of which we can show that a majority lies asymptotically on the Glibert-Varshamov bound, provided the hardness assumption is satisfied[1].

6.1 Basic Notation

We begin with the definitions of the terms we will use throughout the chapter. For simplicity, we restrict ourselves to the particular cases of our interest and will avoid presenting the definitions in full generality. See Appendix A for a quick review of the basic notions in coding theory and [117, 139] for complexity-theoretic notions.

Our main tool in this chapter is a hardness-based pseudorandom generator. Informally, this is an efficient algorithm that receives a sequence of truly random bits at input and outputs a much longer sequence *looking* random to any distinguisher with bounded computational power. This property of the pseudorandom generator can be guaranteed to hold by assuming the existence of functions that are hard to compute for certain computational devices. This is indeed a broad sketch; Depending on what we precisely mean by the quantitative measures just mentioned, we come to different definitions of pseudorandom generators. Here we will be mainly interested in computational hardness against algorithms with *bounded* space complexity.

Hereafter, we will use the shorthand $\mathsf{DSPACE}[s(n)]$ to denote the class of problems solvable with $O(s(n))$ bits of working memory and E for the class of problems solvable in time $2^{O(n)}$ (i.e., $\mathsf{E} = \bigcup_{c\in\mathbb{N}} \mathsf{DTIME}[2^{cn}]$, where $\mathsf{DTIME}[t(n)]$ stands for the class of problems deterministically solvable in time $O(t(n))$).

Certain arguments that we use in this chapter require *non-uniform* computational models. Hence, we will occasionally refer to algorithms that receive *advice strings* to help them carry out their computation. Namely, in addition to the input string, the algorithm receives an *advice* string whose content only depends on the *length* of the input and not the input itself. It is assumed that, for every n, there is an advice string that makes the algorithm work correctly on *all* inputs of length n. We will use the notation $\mathsf{DSPACE}[f(n)]/g(n)$ for the class of problems solvable by algorithms that receive $g(n)$ bits of advice and use $O(f(n))$ bits of working memory.

Definition 6.1. Let $S\colon \mathbb{N} \to \mathbb{N}$ be a (constructible) function. A Boolean function $f\colon \{0,1\}^* \to \{0,1\}$ is said to have *hardness* S if for every algorithm

[1] We remark that the method used in this chapter can be regarded as a "relativized" variation of the original Nisan-Wigderson generator and, apart from construction of error-correcting codes, can be applied to a vast range of probabilistic constructions of combinatorial objects (e.g., Ramsey graphs, combinatorial designs, etc). Even though this derandomization technique seems to be "folklore" among the theoretical computer science community, it is included in the thesis mainly since there appears to be no elaborate and specifically focused writeup of it in the literature.

A in $\mathsf{DSPACE}[S(n)]/O(S(n))$ and infinitely many n (and no matter how the advice string is chosen) it holds that

$$|\Pr_x[A(x) = f(x)] - 1/2| < 1/S(n),$$

where x is uniformly sampled from $\{0,1\}^n$.

Obviously, any Boolean function can be trivially computed correctly on at least half of the inputs by an algorithm that always outputs a constant value (either 0 or 1). Intuitively, for a hard function no *efficient* algorithm can do much better. For the purpose of this chapter, the central hardness assumption that we use is the following:

Assumption 1. *There is a Boolean function in* E *with hardness at least* $2^{\epsilon n}$*, for some constant* $\epsilon > 0$*.*

The term *pseudorandom generator* emphasizes the fact that it is information-theoretically impossible to transform a sequence of truly random bits into a longer sequence of truly random bits, hence the best a transformation with a nontrivial stretch can do is to generate bits that *look* random to a *particular family* of observers. To make this more precise, we need to define *computational indistinguishability* first.

Definition 6.2. Let $p = \{p_n\}$ and $q = \{q_n\}$ be families of probability distributions, where p_n and q_n are distributed over $\{0,1\}^n$. Then p and q are (S, ℓ, ϵ)-*indistinguishable* (for some $S, \ell\colon \mathbb{N} \to \mathbb{N}$ and $\epsilon\colon \mathbb{N} \to (0,1)$) if for every algorithm A in $\mathsf{DSPACE}(S(n))/O(\ell(n))$ and infinitely many n (and no matter how the advice string is chosen) we have that

$$|\Pr_x[A(x) = 1] - \Pr_y[A(y) = 1]| < \epsilon(n),$$

where x and y are sampled from p_n and q_n, respectively.

This is in a way similar to computational hardness. Here the *hard task* is *telling the difference* between the sequences generated by different sources. In other words, two probability distributions are indistinguishable if any resource-bounded observer is *fooled* when given inputs sampled from one distribution rather than the other. Note that this may even hold if the two distributions are not statistically close to each other.

Now we are ready to define pseudorandom generators we will later need.

Definition 6.3. A deterministic algorithm that computes a function

$$G\colon \{0,1\}^{c \log n} \to \{0,1\}^n$$

(for some constant $c > 0$) is called a (high-end) *pseudorandom generator* if the following conditions hold:

1. It runs in polynomial time with respect to n.

2. Let the probability distribution G_n be defined uniformly over the range of G restricted to outputs of length n. Then the family of distributions $\{G_n\}$ is $(n, n, 1/n)$-indistinguishable from the uniform distribution.

An input to the pseudorandom generator is referred to as a *random seed*. Here the length of the output as a function of the seed length s, known as the *stretch* of the pseudorandom generator, is required to be the exponential function $2^{s/c}$.

6.2 The Pseudorandom Generator

A pseudorandom generator, as we just defined, extends a truly random sequence of bits into an exponentially long sequence that *looks* random to any efficient distinguisher. From the definition it is not at all clear whether such an object could exist. In fact the existence of pseudorandom generators (even much weaker than our definition) is not yet known. However, there are various constructions of pseudorandom generators based on unproven (but seemingly plausible) assumptions. The presumed assumption is typically chosen in line with the same guideline, namely, a computational task being *intractable*. For instance, the early constructions of [135] and [15] are based on the intractability of certain number-theoretic problems, namely, integer factorization and the discrete logarithm function. Yao [166] extends these ideas to obtain pseudorandomness from one-way permutations. This is further generalized by [80] who show that the existence of *any* one-way function is sufficient. However, these ideas are mainly motivated by cryptographic applications and often require strong assumptions.

The prototypical pseudorandom generator for the applications in derandomization, which is of our interest, is due to Nisan and Wigderson[115]. They provide a broad range of pseudorandom generators with different strengths based on a variety of hardness assumptions. In rough terms, their generator works by taking a hard function for a certain complexity class, evaluating it in carefully chosen points (related to the choice of the random seed), and outputting the resulting sequence. Then one can argue that an efficient distinguisher can be used to efficiently compute the hard function, contradicting the assumption. Note that for certain complexity classes, hard functions are *provably* known. However, they typically give generators too weak to be applied in typical derandomizations. Here we simply apply the Nisan-Wigderson construction to obtain a pseudorandom generator which is *robust* against space-efficient computations. This is shown in the following theorem:

Theorem 6.4. *Assumption 1 implies the existence of a pseudorandom generator as in Definition 6.3. That is to say, suppose that there is a constant*

$\epsilon > 0$ and a Boolean function computable in time $2^{O(n)}$ that has hardness $2^{\epsilon n}$. Then there exists a function $G\colon \{0,1\}^{O(\log n)} \to \{0,1\}^n$ computable in time polynomial in n whose output (when given uniformly random bits at input) is indistinguishable from the uniform distribution for all algorithms in $\mathsf{DSPACE}[n]/O(n)$.

Proof. [115] Let f be a function satisfying Assumption 1 for some fixed $\epsilon > 0$, and recall that we intend to generate n pseudorandom bits from a truly random seed of length ℓ which is only logarithmically long in n.

The idea of the construction is as follows: We evaluate the hard function f in n carefully chosen points, each of the same length m, where m is to be determined shortly. Each of these m-bit long inputs is obtained from a particular subset of the ℓ bits provided by the random seed. This can be conveniently represented in a matrix form: Let $\mathcal{D}$ be an $n \times \ell$ binary matrix, each row of which having the same weight m. Now the pseudorandom generator G is described as follows: The ith bit generated by G is the evaluation of f on the projection of the ℓ-bit long input sequence to those coordinates indicated by the ith row of $\mathcal{D}$. Note that because f is in E, the output sequence can be computed in time polynomial in n, as long as m is logarithmically small.

As we will shortly see, it turns out that we need $\mathcal{D}$ to satisfy a certain *small-overlap* property. Namely, we require the bitwise product of each pair of the rows of $\mathcal{D}$ to have weight at most $\log n$. A straightforward counting argument shows that, for a logarithmically large value of m, the parameter ℓ can be kept logarithmically small as well. In particular, for the particular choice of $m := \frac{2}{\epsilon} \log n$, the matrix $\mathcal{D}$ exists with $\ell = O(\log n)$. Moreover, rows of the matrix can be constructed (in time polynomial in n) using a simple greedy algorithm.

To show that our construction indeed gives us a pseudorandom generator, suppose that there is an algorithm A working in $\mathsf{DSPACE}[n]/O(n)$ which is able to distinguish the output of G from a truly random sequence with a bias of at least $1/n$. That is, for all large enough n it holds that

$$\delta := |\Pr_y[A^{\alpha(n)}(y) = 1] - \Pr_x[A^{\alpha(n)}(G(x)) = 1]| \geq 1/n,$$

where x and y are distributed uniformly in $\{0,1\}^\ell$ and $\{0,1\}^n$, respectively, and $\alpha(n)$ in the superscript denotes an advice string of linear length (that only depends on n). The goal is to transform A into a space-efficient (and non-uniform) algorithm that approximates f, obtaining a contradiction.

Without loss of generality, let the quantity inside the absolute value be non-negative (the argument is similar for the negative case). Let the distribution D_i (for $0 \leq i \leq n$) over $\{0,1\}^n$ be defined by concatenation of the length-i prefix of $G(x)$, when x is chosen uniformly at random from $\{0,1\}^\ell$, with a Boolean string of length $n - i$ obtained uniformly at random. Define p_i as $\Pr_z[A^{\alpha(n)}(z) = 1]$, where z is sampled from D_i, and let $\delta_i := p_{i-1} - p_i$. Note

that D_0 is the uniform distribution and D_n is uniformly distributed over the range of G. Hence, we have $\sum_{i=1}^{n} \delta_i = p_0 - p_n = \delta \geq 1/n$, meaning that for some i, $\delta_i \geq 1/n^2$. Fix this i in the sequel.

Without loss of generality, assume that the ith bit of $G(x)$ depends on the first m bits of the random seed. Now consider the following randomized procedure B: Given $i-1$ input bits $u_1, \ldots, u_{i-1}$, choose a binary sequence $r_i, \ldots, r_n$ uniformly at random and compute $A^{\alpha(n)}(u_1, \ldots, u_{i-1}, r_i, \ldots, r_n)$. If the output was 1 return r_i, otherwise, return the negation of r_i. It is straightforward to show that

$$\Pr_{x,r}[B(G(x)_1^{i-1}) = G(x)_i] \geq \frac{1}{2} + \delta_i. \tag{6.1}$$

Here, $G(x)_1^{i-1}$ and $G(x)_i$ are shorthands for the $(i-1)$-bit long prefix of $G(x)$ and the ith bit of $G(x)$, respectively, and the probability is taken over the choice of x and the internal coins of B.

So far we have constructed a linear-time probabilistic procedure for *guessing* the ith pseudorandom bit from the first $i-1$ bits. By averaging, we note that there is a particular choice of $r_i, \ldots, r_n$, independent of x, that preserves the bias given in (6.1). Furthermore, note that the function $G(x)_i$ we are trying to guess, which is in fact $f(x_1, \ldots, x_m)$, does *not* depend on $x_{m+1}, \ldots, x_\ell$. Therefore, again by averaging we see that these bits can also be fixed. Therefore, for a given sequence $x_1, \ldots, x_m$, one can compute $G(x)_1^{i-1}$, feed it to B (having known the choices we have fixed), and guess $G(x)_i$ with the same bias as in (6.1). The problem is of course that $G(x)_1^{i-1}$ does not seem to be easily computable. However, what we know is that each bit of this sequence depends only on $\log n$ bits of $x_1, \ldots, x_m$, followed by the construction of $\mathcal{D}$. Hence, having fixed $x_{m+1}, \ldots, x_\ell$, we can trivially describe each bit of $G(x)_1^{i-1}$ by a Boolean formula (or a Boolean circuit) of exponential size (that is, of size $O(2^{\log n}) = O(n)$). These $i - 1 = O(n)$ Boolean formulae can be encoded as an additional advice string of length $O(n^2)$ (note that their descriptions only depend on n), implying that $G(x)_1^{i-1}$ can be computed in linear space using $O(n^2)$ bits of advice.

All the choices we have fixed so far (namely, $i, r_i, \ldots, r_n, x_{m+1}, \ldots, x_\ell$) only depend on n and can be *absorbed* into the advice string as well[2]. Combined with the *bit-guessing* algorithm we just described, this gives us a linear-space algorithm that needs an advice of quadratic length and correctly computes $f(x_1, \ldots, x_m)$ on at least a $\frac{1}{2} + \delta_i$ fraction of inputs, which is off from $1/2$ by a bias of at least $1/n^2$. But this is not possible by the hardness of f, which is assumed to be at least $2^{\epsilon m} = n^2$. Thus, G must be a pseudorandom generator. □

[2] Alternatively, one can avoid using this additional advice by enumerating over all possible choices and taking a majority vote. However, this does not decrease the total advice length by much.

The above proof uses a function that is completely unpredictable for every efficient algorithm. Impagliazzo and Wigderson [85] improve the construction to show that this requirement can be relaxed to one that only requires a *worst case* hardness, meaning that the function computed by any *efficient* (non-uniform) algorithm needs to differ from the hard function on at least one input. In our application, this translates into the following hardness assumption:

Assumption 2. *There is a constant $\epsilon > 0$ and a function f in E such that every algorithm in $\mathsf{DSPACE}[S(n)]/O(S(n))$ that correctly computes f requires $S(n) = \Omega(2^{\epsilon n})$.*

The idea of their result (which was later reproved in [146] using a coding-theoretic argument) is to *amplify* the given hardness, that is, to transform a worst-case hard function in E to another function in E which is hard on average. In our setting, this gives us the following (since the proof essentially carries over without change, we only sketch the idea):

Theorem 6.5. *Assumption 2 implies Assumption 1 and hence, the existence of pseudorandom generators.*

Proof Idea. [146] Let a function f be hard in worst case. Consider the truth table of f as a string x of length $N := 2^n$. The main ingredient of the proof is a linear code $\mathcal{C}$ with dimension N and length polynomial in N, which is obtained by concatenation of a Reed-Muller code with the Hadamard code. The code is list-decodable up to a fraction $\frac{1}{2} - \epsilon$ of errors, for arbitrary $\epsilon > 0$. Moreover, decoding can be done in sub-linear time, that is, by querying the *received word* only at a small number of (randomly chosen) positions. Then the truth table of the transformed function g can be simply defined as the encoding of x with $\mathcal{C}$. Hence g can be evaluated at any point in time polynomial in N, which shows that $g \in \mathsf{E}$. Further, suppose that an algorithm A can space-efficiently compute g correctly in a fraction of points non-negligibly bounded away from $1/2$ (possibly using an advice string). Then the function computed by A can be seen as a *corrupted* version of the *codeword* g and can be efficiently *recovered* using the list-decoding algorithm. From this, one can obtain a space-efficient algorithm for computing f, contradicting the hardness of f. Hence g has to be hard on average. □

While the above result seems to require hardness against non-uniform algorithms (as phrased in Assumption 2), we will see that the hardness assumption can be further relaxed to the following, which only requires hardness against uniform algorithms:

Assumption 3. *The complexity class E is not contained in $\mathsf{DSPACE}[2^{o(n)}]$.*

Remark. A result by Hopcroft et al. [83] shows a deterministic simulation of time by space. Namely, they prove that

$$\mathsf{DTIME}[t(n)] \subseteq \mathsf{DSPACE}[t(n)/\log t(n)].$$

However, this result is not strong enough to influence the hardness assumption above. To violate the assumption, a much more space-efficient simulation in the form

$$\mathsf{DTIME}[t(n)] \subseteq \mathsf{DSPACE}[t(n)^{o(1)}]$$

is required.

Before we show the equivalence of the two assumptions (namely, Assumption 2 and Assumption 3), we address the natural question of how to construct an *explicit* function to satisfy the required hardness assumption (after all, evaluation of such a function is needed as part of the pseudorandom generator construction). One possible candidate (which is a canonical hard function for E) is proposed in the following lemma:

Lemma 6.6. *Let $\mathcal{L}_{\mathsf{E}}$ be the set (encoded in binary)*

$$\{\langle M, x, t, i\rangle \mid M \textit{ is a Turing machine, where given input } x \textit{ at time } t \textit{ the } i\textit{th bit of its configuration is } 1\},$$

and let the Boolean function f_{E} be its characteristic function. Then if Assumption 3 is true, it is satisfied by f_{E}.

Proof. First we show that $\mathcal{L}_{\mathsf{E}}$ is complete for E under Turing reductions bounded in linear space. The language being in E directly follows from the efficient constructions of universal Turing machines. Namely, given a properly-encoded input $\langle M, x, t, i\rangle$, one can simply simulate the Turing machine M on x for t steps and decide according to the configuration obtained at time t. This indeed takes exponential time. Now let L be any language in E which is computable by a Turing machine M in time 2^{cn}, for some constant $c > 0$. For a given x of length n, using an oracle for solving f_{E}, one can query the oracle with inputs of the form $\langle M, x, 2^{cn}, i\rangle$ (where the precise choice of i depends on the particular encoding of the configurations) to find out whether M is in an accepting state, and hence decide L. This can obviously be done in space linear in n, which concludes the completeness of $\mathcal{L}_{\mathsf{E}}$. Now if Assumption 3 is true and is not satisfied by f_{E}, this completeness result allows one to compute all problems in E in sub-exponential time, which contradicts the assumption. □

The following lemma shows that this seemingly weaker assumption is in fact sufficient for our pseudorandom generator:

Lemma 6.7. *Assumptions 2 and 3 are equivalent.*

Proof. This argument is based on [108, Section 5.3]. First we observe that, given a *black box* C that receives n input bits and outputs a single bit, it can be verified in linear space whether C computes the restriction of f_{E} to inputs of length n. To see this, consider an input of the form $\langle M, x, t, i\rangle$, as in

the statement of Lemma 6.6. The correctness of C can be explicitly checked when the time parameter t is zero (that is, C has to agree with the initial configuration of M). Moreover, for every time step $t > 0$, the answer given by C has to be consistent with that of the previous time step (namely, the transition made at the location of the head should be legal and every other position of the tape should remain unchanged). Thus, on can verify C simply by enumerating all possible inputs and verifying whether the answer given by C remains consistent across subsequent time steps. This can obviously be done in linear space.

Now suppose that Assumption 3 is true and hence, by Lemma 6.6, is satisfied by f_{E}. That is, there is a constant $\epsilon > 0$ such that every algorithm for computing f_{E} requires space $O(2^{\epsilon n})$. Moreover, assume that there is an algorithm A working in $\mathsf{DSPACE}[S(n)]/O(S(n))$ that computes f_{E}. Using the verification procedure described above, one can (uniformly) simulate A in space $O(S(n))$ by enumerating all choices of the advice string and finding the one that makes the algorithm work correctly. Altogether this requires space $O(S(n))$. Combined with the hardness assumption, we conclude that $S(n) = \Omega(2^{\epsilon n})$. The converse direction is obvious. □

Putting everything together, we obtain a very strong pseudorandom generator as follows:

Corollary 6.8. *Assumption 3 implies the existence of pseudorandom generators whose output of length n is $(n, n, 1/n)$-indistinguishable from the uniform distribution.* □

6.3 Derandomized Code Construction

As mentioned before, the bound given by Gilbert and Varshamov[68, 157] states that, for a q-ary alphabet, large enough n, and for any value of $0 \leq \delta \leq (q-1)/q$, there are codes with length n, relative distance at least δ and rate $r \geq 1 - h_q(\delta)$, where h_q is the q-ary entropy function. Moreover, a random linear code (having each entry of its generator matrix chosen uniformly at random) achieves this bound. In fact, for all $0 \leq r \leq 1$, in the family of linear codes with length n and (designed) dimension nr, all but only a sub-constant fraction of the codes achieve the bound when n grows to infinity. However, the number of codes in the family is exponentially large (q^{nr}) and we do not have an *a priori* indication on which codes in the family are good. Putting it differently, a randomized algorithm that merely outputs a random generator matrix *succeeds* in producing a code on the GV bound with probability $1-o(1)$. However, the number of random bits needed by the algorithm is $nk \log q$. For simplicity, in the sequel we only focus on binary codes, for which no explicit construction approaching the GV bound is known.

The randomized procedure above can be considerably derandomized by considering a more restricted family of codes. Namely, fix a length n and a basis for the finite field $\mathbb{F}_m$, where $m := 2^{n/2}$. Then over such a basis there is a natural isomorphism between the elements of $\mathbb{F}_m$ and the elements of the vector space $\mathbb{F}_2^{n/2}$. Now for each $\alpha \in \mathbb{F}_m$, define the code $\mathcal{C}_\alpha$ as the set $\{\langle x, \alpha x\rangle \mid x \in \mathbb{F}_m\}$, where the elements are encoded in binary[3]. This binary code has rate $1/2$. Further, it is well known that $\mathcal{C}_\alpha$ achieves the GV bound for all but $1 - o(1)$ fraction of the choices of α. Hence in this family a randomized construction can obtain very good codes using only $n/2$ random bits. Here we see how the pseudorandom generator constructed in the last section can dramatically reduce the amount of randomness needed in all code constructions. Our observation is based on the composition of the following facts:

1. *Random codes achieve the Gilbert-Varshamov bound:* It is well known that a simple randomized algorithm that chooses the entries of a generator matrix uniformly at random obtains a linear code satisfying the Gilbert-Varshamov bound with overwhelming probability [157].

2. *Finding the minimum distance of a (linear) code can be performed in linear space:* One can simply enumerate all the codewords to find the minimum weight codeword, and hence, the distance of the code. This only requires linear amount of memory with respect to the block length.

3. *Provided a* hardness condition*, namely that sub-exponential space algorithms cannot compute all the problems in* E*, every linear space algorithm can be* fooled *by an explicit pseudorandom generator:* This is what we obtained in Corollary 6.8.

Now we formally propose a general framework that can be employed to derandomize a wide range of combinatorial constructions.

Lemma 6.9. *Let $\mathcal{S}$ be a family of combinatorial objects of (binary-encoded) length n, in which an ϵ fraction of the objects satisfy a property P. Moreover, suppose that the family is efficiently samplable, that is, there is a polynomial-time algorithm (in n) that, for a given i, generates the ith member of the family. Further assume that the property P is verifiable in polynomial space. Then for every constant $k > 0$, under Assumption 3, there is a constant ℓ and an efficiently samplable subset of $\mathcal{S}$ of size at most n^ℓ in which at least an $\epsilon - n^{-k}$ fraction of the objects satisfy P.*

Proof. Let A be the composition of the sampling algorithm with the verifier for P. By assumption, A needs space n^s, for some constant s. Furthermore, when the input of A is chosen randomly, it outputs 1 with probability at least ϵ.

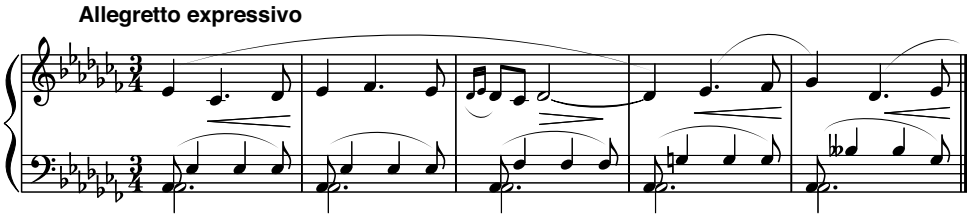
[3]These codes are attributed to J. M. Wozencraft (see [106]).

Suppose that the pseudorandom generator of Corollary 6.8 transforms $c \log n$ truly random bits into n pseudorandom bits, for some constant $c > 0$. Now it is just enough to apply the pseudorandom generator on $c \cdot \max\{s, k\} \cdot \log n$ random bits and feed n of the resulting pseudorandom bits to A. By this construction, when the input of the pseudorandom generator is chosen uniformly at random, A must still output 1 with probability $\epsilon - n^{-k}$ as otherwise the pseudorandomness assumption would be violated. Now the combination of the pseudorandom generator and A gives the efficiently samplable family of the objects we want, for $\ell := c \cdot \max\{s, k\}$, as the random seed runs over all the possibilities. □

As the distance of a code is obviously computable in linear space by enumeration of all the codewords, the above lemma immediately implies the existence of a (constructible) polynomially large family of codes in which at least $1 - n^{-k}$ of the codes achieve the GV bound, for arbitrary k.

Remark. As shown in the original work of Nisan and Wigderson [115] (followed by the hardness amplification of Impagliazzo and Wigderson [85]) all randomized polynomial-time algorithms (namely, the complexity class BPP) can be fully derandomized under the assumption that E cannot be computed by Boolean circuits of sub-exponential size. This assumption is also sufficient to derandomize probabilistic constructions that allow a (possibly non-uniform) polynomial-time verification procedure for deciding whether a particular object has the desirable properties. For the case of good error-correcting codes, this could work if we knew of a procedure for computing the minimum distance of a linear code using circuits of size polynomial in the length of the code. However, it turns out that (the decision version of) this problem is NP-complete [156], and even the approximation version remains NP-complete [52]. This makes such a possibility unlikely.

However, a key observation, due to Klivans and van Melkebeek [92], shows that the Nisan-Wigderson construction (as well as the Impagliazzo-Wigderson amplification) can be *relativized*. Namely, starting from a hardness assumption for a certain family of *oracle circuits* (i.e., Boolean circuits that can use special gates to compute certain Boolean functions as *black box*) one can obtain pseudorandom generators secure against oracle circuits of the same family. In particular, this implies that any probabilistic construction that allows polynomial time verification using NP oracles (including the construction of good error-correcting codes) can be derandomized by assuming that E cannot be computed by sub-exponential sized Boolean circuits that use NP oracle gates. However, the result given by Lemma 6.9 can be used to derandomize a more general family of probabilistic constructions, though it needs a slightly stronger hardness assumption which is still plausible.



Isaac Albéniz (1860–1909): Iberia Suite for Piano, Book 1, Evocación in A flat.

"To achieve great things, two things are needed; a plan, and not quite enough time."

— *Leonard Bernstein*

Chapter 7

Concluding Remarks

In this thesis, we investigated the role of objects studied at the core of theoretical computer science–namely, randomness extractors, condensers and pseudorandom generators–in efficient construction of combinatorial objects suitable for more practical applications. The applications being considered all share a coding-theoretic flavor and include:

1. Wiretap coding schemes, where the goal is to provide information-theoretic secrecy in a communication channel that is partially observable by an adversary (Chapter 3);

2. Combinatorial group testing schemes, that allow for efficient identification of sparse binary vectors using potentially unreliable disjunctive measurements (Chapter 4);

3. Capacity achieving codes, which provide optimally efficient and reliable transmission of information over unreliable discrete communication channels (Chapter 5);

4. Codes on the Gilbert-Varshamov bound, which are error-correcting codes whose rate-distance trade-off matches what achieved by probabilistic constructions (Chapter 6).

We conclude the thesis by a brief and informal discussion of the obtained results, open problems and possible directions for future research.

Wiretap Protocols

In Chapter 3 we constructed rate-optimal wiretap schemes from optimal affine extractors. The combinatorial structure of affine extractors guarantees almost perfect privacy even in presence of linear manipulation of information. This

observation was the key for our constructions of information-theoretically optimal schemes in presence of noisy channels, active intruders, and linear network coding.

Despite being sufficiently general for a wide range of practical applications, it makes sense to consider different types of intermediate processing. We showed in Section 3.7.3 that, at the cost of giving up zero leakage, it is possible to use seeded extractors to provide secrecy in presence of arbitrary forms of transformations. However, in order to attain zero leakage, it becomes inevitable to construct seedless, invertible extractors for a class of random sources that capture the nature of post-processing being allowed.

For example, suppose that the encoded information is transmitted through a packet network towards a destination, where information is arbitrarily manipulated by intermediate routers, but is routed from the source to the destination through $k \geq 2$ separated paths. In this case, the intruder may learn a limited amount of information from each of the k components of the network. Similar arguments as what presented in Chapter 3 can now be used to show that the object needed for ensuring secrecy in this "route-disjoint" setting is invertible, k-source extractors. Shaltiel [132] demonstrates that his method for boosting the output size of extractors using output-length optimal seeded extractors (that is the basis of our technique for making seedless extractors invertible) can be extended to the case of two-source extractors as well.

On the other hand, if the route-disjointness condition that is assumed in the above example is not available, zero leakage can no longer be guaranteed without imposing further restrictions (since, as discussed in Section 3.7.3, this would require seedless extractors for general sources, which do not exist). However, assume that the intermediate manipulations are carried out by computationally bounded devices (a reasonable assumption to model the real world). A natural candidate for modeling resource-bounded computation is the notion of small-sized Boolean circuits. The secrecy problem for this class of transformations leads to invertible extractors for the following class of sources:

> For an arbitrary Boolean function $C\colon \{0,1\}^n \to \{0,1\}$ that is computable by Boolean circuits of bounded size, the source is uniformly distributed on the set of inputs $x \in \{0,1\}^n$ such that $C(x) = 0$ (assuming that this set has a sufficiently large size).

In a recent work of Shaltiel [133], this type of extractors have been studied under the notion of "extractors for *recognizable* sources" (a notion that can be specialized to different sub-classes depending on the bounded model of computation being considered.

On the other hand, Trevisan and Vadhan [153] introduce the related notion of extractors for *samplable sources*, where a samplable source is defined as the image of a small-sized circuit (having multiple outputs) when provided with a

uniformly random input. They proceed to show explicit constructions of such extractors assuming suitable computational hardness assumptions (which turn out to be to some extent necessary for such extractors to be constructible). It is straightforward to see that their techniques can be readily extended to construction of explicit extractors for sources recognizable by small-sized circuits (using even weaker hardness assumptions). However, the technique works when the source entropy is assured to be substantially large, and even so, is unable to produce a nearly optimal output length. To this date, explicit construction of better extractors, under mild computational assumptions, for sources that are samplable (or recognizable) by small-sized circuits remains an important open problem.

Observe that the technique of using extractors for construction of wiretap protocols as presented in Chapter 3 achieves optimal rates only if the wiretap channel (i.e., the channel that delivers intruder's information) is of *erasure* nature. That is, we have so far assumed that, after some possible post-processing of the encoded information, the intruder observes an arbitrarily chosen, but bounded, subset of the bits being transmitted and remains unaware of the rest. There are different natural choices of the wiretap channel that can be considered as well. For example, suppose that the intruder observes a *noisy* version of the entire sequence being transmitted (e.g., when a fraction of the encoded bits get randomly flipped before being delivered to the intruder). An interesting question is to see whether invertible extractors (or a suitable related notion) can be used to construct information-theoretically optimal schemes for such variations as well.

Group Testing

Non-adaptive group testing schemes are fundamental combinatorial objects of both theoretical and practical interest. As we showed in Chapter 4, strong condensers can be used as building blocks in construction of noise-resilient group testing and threshold group testing schemes.

The factors that greatly influence the quality of our constructions are the seed length and output length of the condenser being used. As we saw, in order to obtain an asymptotically optimal number of measurements, we need explicit constructions of extractors and lossless condensers that achieve a logarithmic seed length, and output length that is different from the source entropy by small additive terms. While, as we saw, there are very good existing constructions of both extractors and lossless condensers that can be used, they are still sub-optimal in the above sense. Thus, any improvement on the state of the art in explicit construction of extractors and lossless condensers will immediately improve the qualities of our explicit constructions.

Moreover, our constructions of noise-resilient schemes with sublinear decoding time demonstrates a novel application for list-decodable extractors and

condensers. This motivates further investigation of these objects for improvement of their qualities.

In Section 4.3, we introduced the combinatorial notion of $(d, e; u)$-regular matrices, that is used as an intermediate tool towards obtaining threshold testing designs. Even though our construction, assuming an optimal lossless condenser, matches the probabilistic upper bound for regular matrices, the number of measurements in the resulting threshold testing scheme will be larger than the probabilistic upper bound by a factor of $\Omega(d \log n)$. Thus, an outstanding question is coming up with a direct construction of disjunct matrices that match the probabilistic upper bound.

Despite this, the notion of regular matrices may be of independent interest, and an interesting question is to obtain (nontrivial) concrete lower bounds on the number of rows of such matrices in terms of the parameters d, e, u.

Moreover, in our constructions we have assumed the threshold u to be a fixed constant, allowing the constants hidden in asymptotic notions to have a poor dependence on u. An outstanding question is whether the number of measurements can be reasonably controlled when u becomes large; e.g., $u = \Omega(d)$.

Another interesting problem is decoding in the threshold model. While our constructions can combinatorially guarantee identification of sparse vectors, for applications it is important to have an efficient reconstruction algorithm as well. Contrary to the case of strongly disjunct matrices that allow a straightforward decoding procedure (cf. [27]), it is not clear whether in general our notion of disjunct matrices allow efficient decoding, and thus it becomes important to look for constructions that are equipped with efficient reconstruction algorithms.

Finally, for clarity of the exposition, in this presentation we have only focused on asymptotic trade-offs, and it would be nice to obtain good, non-asymptotic, estimates on the obtained bounds that are useful for applications.

Capacity Achieving Codes

The general construction of capacity-achieving codes presented in Chapter 5 can be used to obtain a polynomial-sized ensemble of codes of any given block length n, provided that nearly optimal linear extractors or lossless condensers are available. In particular, this would require a logarithmic seed length and an output length which is different from the input entropy by an arbitrarily small constant fraction of the entropy. Both extractors and lossless condensers constructed by Guruswami, Umans, and Vadhan [78] achieve this goal, and as we saw in Chapter 2, their lossless condenser can be easily made linear. However, to the best of our knowledge, to this date no explicit construction of a linear extractor with logarithmic seed length that extracts even a constant fraction of the source entropy is known.

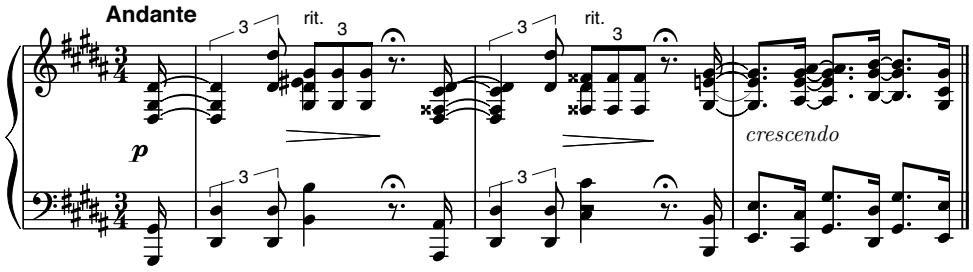
Another interesting problem concerns the duality principle presented in Section 5.5. As we showed, linear affine extractors and lossless condensers are dual objects. It would be interesting to see whether a more general duality principle exist between extractors and lossless condensers. It is not hard to use basic Fourier analysis to slightly generalize our result to linear extractors and lossless condensers for more general (not necessarily affine) sources. However, since condensers for general sources are allowed to have a positive, but negligible error (which is *not* the case for linear affine condensers), controlling the error to a reasonable level becomes a tricky task, and forms an interesting problem for future research.

The Gilbert-Varshamov Bound

As we saw in Chapter 6, a suitable computational assumption implies a deterministic polynomial-time algorithm for explicit construction of polynomially many linear codes of a given length n, almost all of which attaining the Gilbert-Varshamov bound. That is, a randomly chosen code from such a short list essentially behaves like a fully random code and in particular, is expected to attain the same rate-distance tradeoff.

An important question that remains unanswered is whether a single code of length n attaining the Gilbert-Varshamov bound can be efficiently constructed from a list of $\mathsf{poly}(n)$ codes in which an overwhelming fraction attain the bound. In effect, we are looking for an efficient *code product* to combine a polynomially long list of codes (that may contain a few *unsatisfactory* codes) into a single code that possesses the qualities of the overwhelming majority of the codes in the ensemble. Since the computational problem of determining (or even approximating) the minimum distance of a linear code is known to be intractable, such a product cannot be constructed by simply examining the individual codes. It is also interesting to consider impossibility results, that is, models under which such a code product may become as difficult to construct as finding a good code "from scratch".

Finally, a challenging problem which still remains open is explicit construction of codes (or even small ensembles of codes) that attain the Gilbert-Varshamov bound without relying on unproven assumptions. For sufficiently large alphabets (i.e., of size 49 or higher), geometric Goppa codes are known to even surpass the GV bound [154]. However, for smaller alphabets, or rates close to zero over constant-sized alphabets, no explicit construction attaining the GV bound is known. It also remains unclear whether in such cases the GV bound is optimal; that is, whether there are families of codes, not necessarily explicit, that beat the bound.



Alexander Scriabin (1872–1915): Piano Sonata No. 2 in G sharp minor (Op. 19, "Sonata-Fantasy").

"A classic is a book that has never finished saying what it has to say."

— *Italo Calvino*

Appendix A

A Primer on Coding Theory

In this appendix, we briefly overview the essential notions of coding theory that we have used in the thesis. For an extensive treatment of the theory of error-correcting codes (an in particular, the facts collected in this appendix), we refer the reader to the books by MacWilliams and Sloane [103], van Lint [98], and Roth [127] on the topic.

A.1 Basics

Let Σ be a finite alphabet of size $q > 1$. A code $\mathcal{C}$ of length n over Σ is a non-empty subset of Σ^n. Each element of $\mathcal{C}$ is called a *codeword* and $|\mathcal{C}|$ defines the *size* of the code. The *rate* of the $\mathcal{C}$ is defined as $\log_q |\mathcal{C}|/n$. An important choice for the alphabet is $\Sigma = \{0, 1\}$, which results in a *binary code*. Typically, we assume that q is a prime power and take Σ to be the finite field $\mathbb{F}_q$.

The Hamming distance between vectors $w := (w_1, \ldots, w_n) \in \Sigma^n$ and $w' := (w'_1, \ldots, w'_n) \in \Sigma^n$ is defined as the number of positions at which w and w' differ. Namely,

$$\mathsf{dist}(w, w') := |\{i \in [n] \colon w_i \neq w'_i\}|.$$

The *Hamming weight* of a vector $w \in \mathbb{F}_q^n$ (denoted by $\mathsf{wgt}(w)$) is the number of its nonzero coordinates; i.e.,

$$\mathsf{wgt}(w) := |\{i \in [n] \colon w_i \neq 0\}|.$$

Therefore, when $w, w' \in \mathbb{F}_q^n$, we have

$$\mathsf{dist}(w, w') = \mathsf{wgt}(w - w').$$

The *minimum distance* of a code $\mathcal{C} \subseteq \Sigma^n$ is the quantity

$$\mathsf{dist}(\mathcal{C}) := \min_{w, w' \in C} \mathsf{dist}(w, w'),$$

and the *relative distance* of the code is defined as $\mathsf{dist}(\mathcal{C})/n$. A family of codes of growing block lengths n is called *asymptotically good* if, for large enough n, it achieves a positive constant rate (i.e., independent of n) and a positive constant relative distance.

A code $\mathcal{C} \in \mathbb{F}_q^n$ is called *linear* if it is a vector subspace of $\mathbb{F}_q^n$. In this case, the *dimension* of the code is defined as its dimension as a subspace, and the rate would be given by the dimension divided by n. A code $\mathcal{C}$ with minimum distance d is denoted by the shorthand $(n, \log_q |\mathcal{C}|, d)_q$, and when $\mathcal{C}$ is linear with dimension k, by $[n, k, d]_q$. The subscript q is omitted for binary codes. Any linear code must include the all-zeros word 0^n. Moreover, due to the linear structure of such codes, the minimum distance of a linear code is equal to the minimum Hamming weight of its nonzero codewords.

A *generator matrix* G for a linear $[n, k, d]_q$-code $\mathcal{C}$ is a $k \times n$ matrix of rank k over $\mathbb{F}_q$ such that

$$\mathcal{C} = \{xG \colon x \in \mathbb{F}_q^k\}.$$

Moreover, a *parity check matrix* H for $\mathcal{C}$ is an $r \times n$ matrix over $\mathbb{F}_q$ of rank $n - k$, for some $r \geq n - k$, such that[1]

$$\mathcal{C} = \{x \in \mathbb{F}_q^n \colon Hx^\top = 0\}.$$

Any two such matrices are orthogonal to one another, in that we must have $GH^\top = 0$. It is easy to verify that, if $\mathcal{C}$ has minimum distance d, then every choice of up to $d - 1$ columns of H are linearly independent, and there is a set of d columns of H that are dependent (and the dependency is given by a codeword of minimum weight).

The *dual* of a linear code $\mathcal{C}$ of length n over $\mathbb{F}_q$ (denoted by $\mathcal{C}^\top$) is defined as the dual vector space of the code; i.e., the set of vectors in $\mathbb{F}_q^n$ that are all orthogonal to every codeword in $\mathcal{C}$:

$$\mathcal{C}^\perp := \{c \in \mathbb{F}_q^n \colon (\forall w \in \mathcal{C})\ c \cdot w^\top = 0\}.$$

The dual of a k-dimensional code has dimension $n - k$, and $(\mathcal{C}^\perp)^\perp = \mathcal{C}$. Moreover, a generator matrix for the code $\mathcal{C}$ is a parity check matrix for $\mathcal{C}^\perp$ and vice versa.

An *encoder* for a code $\mathcal{C}$ with q^k codewords is a function $E \colon \Sigma^k \to \Sigma^n$ whose image is the code $\mathcal{C}$. In particular, this means that E must be injective (one-to-one). Moreover, any generator matrix for a linear code defines the encoder $E(x) := xG$. The input x is referred to as the *message*. We will consider a code *explicit* if it is equipped with a polynomial-time computable encoder. For linear codes, this is equivalent to saying that there is a deterministic polynomial time algorithm (in the length n) that outputs a generator, or parity check, matrix for the code[2].

[1] Here we consider vectors as row vectors, and denote column vectors (e.g., $x^\top$) as transpose of row vectors.

[2] There are more strict possibilities for considering a code explicit; e.g., one may require each entry of a generator matrix to be computable in logarithmic space.

Given a message $x \in \Sigma^k$, assume that an encoding of x is obtained using an encoder E; i.e., $y := E(x) \in \Sigma^n$. Consider a *communication channel* through which the encoded sequence y is communicated. The output of the channel $\tilde{y} \in \Sigma^n$ is delivered to a receiver, whose goal is to reconstruct x from $\tilde{y}$. Ideally, if the channel is perfect, we will $\tilde{y} = y$ and, since $E(x)$ is injective, deducing x amounts to inverting the function E, which is an easy task for linear codes (in general, this can be done using Gaussian elimination). However, consider a *closest distance* decoder $D\colon \Sigma^n \to \Sigma^k$ that, given $\tilde{y}$, outputs an $x \in \Sigma^k$ for which $\mathsf{dist}(E(x), \tilde{y})$ is minimized. It is easy to see that, even if we allow the channel to arbitrarily alter up to $t := \lfloor (d-1)/2 \rfloor$ of the symbols in the transmitted sequence y (in symbols, if $\mathsf{dist}(y, \tilde{y}) \leq t$), then we can still ensure that x is uniquely deducible from $\tilde{y}$; in particular, we must have $D(\tilde{y}) = x$.

For a linear code over $\mathbb{F}_q$ with parity check matrix H, a *syndrome* corresponding to a sequence $\tilde{y} \in \mathbb{F}_q^n$ is the vector $H\tilde{y}^\top$. Thus, $\tilde{y}$ is a codeword if and only if its corresponding syndrome is the zero vector. Therefore, in the channel model above, if the syndrome corresponding to the received word $\tilde{y}$ is nonzero, we can be certain that $\tilde{y} \neq y$. The converse is not necessarily true. However, it is a simple exercise to see that if $\tilde{y}$ and $\tilde{y}' \in \mathbb{F}_q^n$ are both such that $\tilde{y} \neq \tilde{y}'$ and moreover $\mathsf{dist}(y, \tilde{y}) \leq t$ and $\mathsf{dist}(y, \tilde{y}) \leq t$, then the corresponding syndromes must be different; i.e., $H\tilde{y}^\top \neq H\tilde{y}'^\top$. Therefore, provided that the number of errors is no more than the "unique-decoding threshold" t, it is "combinatorially" possible to uniquely reconstruct x from the syndrome corresponding to the received word. This task is known as *syndrome decoding*. However, ideally it is desirable to have an efficient algorithm for syndrome decoding as well that runs in polynomial time in the length of the code. In general, syndrome decoding for a linear code defined by its parity check matrix is NP-hard (see [12]). However, a variety of explicit code constructions are equipped with efficient syndrome decoding algorithms.

As discussed above, a code with minimum distance d can tolerate up to $t := \lfloor (d-1)/2 \rfloor$ errors. Moreover, if the number of errors can potentially be larger than t, then a confusion becomes unavoidable and unique decoding can no longer be guaranteed. However, the notion *list decoding* allows to control the "amount of confusion" when the number of errors is more than t. Namely, for a *radius* ρ and integer ℓ (referred to as the *list size*), a code $\mathcal{C} \subseteq [q]^n$ is called (ρ, ℓ) list-decodable if the number of codewords within a distance ρn of any vector in $[q]^n$ is at most ℓ. In this view, unique decoding corresponds to the case $\ell = 1$, and a code with minimum distance d is $(\frac{1}{n}\lfloor (d-1)/2 \rfloor, 1)$ list-decodable. However, for many theoretical and practical purposes, a small (but possibly much larger than 1) list size may be sufficient.

A.2 Bounds on codes

For positive integers n, d, q, denote by $A_q(n, d)$ the maximum size of a code with length n and minimum distance d over a q-ary alphabet, and define

$$\alpha_q(\delta) := \lim_{n\to\infty} \frac{\log_q A(n, \delta n)}{n}$$

as the "highest" rate a code with relative distance δ can asymptotically attain. The exact form of the function $\alpha_q(\cdot)$ is not known for any q; however, certain lower and upper bounds for this quantity exist. In this section, we briefly review some important bounds on $\alpha_q(\delta)$.

The Gilbert-Varshamov bound

Using the probabilistic method, it can be shown that a random linear code (constructed by picking the entries of its generator, or parity check, matrix uniformly and independently at random) with overwhelming probability attains a dimension-distance tradeoff given by

$$k \geq n(1 - h_q(d/n)),$$

where $h_q(\cdot)$ is the q-ary entropy function defined as

$$h_q(x) := x \log_q(q-1) - x \log_q(x) - (1-x)\log_q(1-x). \tag{A.1}$$

Thus we get the lower bound

$$\alpha_q(\delta) \geq 1 - h_q(\delta)$$

on the function $\alpha_q(\cdot)$, known as the Gilbert-Varshamov bound.

The Singleton bound

On the negative side, the Singleton bound states that the minimum distance d of any q-ary code with q^k or more codewords must satisfy $d \leq n - k + 1$. Codes that attain this bound with equality are known as *maximum distance separable (MDS)* codes. Therefore we get that, regardless of the alphabet size, one must have

$$\alpha_q(\delta) \leq 1 - \delta.$$

Lower bounds for fixed alphabet size

When the alphabet size q is fixed, there are numerous lower bounds known for the function $\alpha_q(\cdot)$. Here we list several such bounds.

- *Hamming (sphere packing) bound:* $\alpha_q(\delta) \leq 1 - h_q(\delta/2)$.

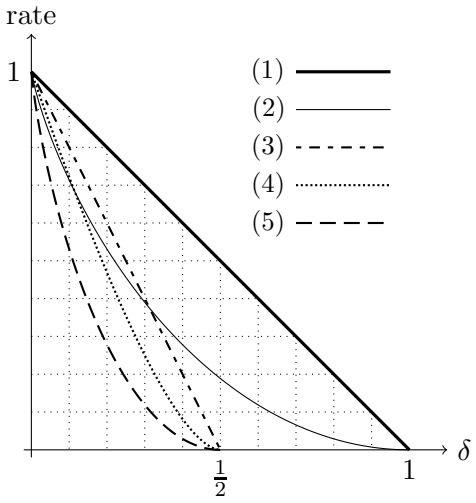


Figure A.1: Bounds on binary codes: (1) Singleton bound, (2) Hamming bound, (3) Plotkin bound, (4) MRRW bound, (5) Gilbert-Varshamov bound.

- *Plotkin bound:* $\alpha_q(\delta) \leq \max\{0, 1 - \delta(q/(q-1))\}$.
- *McEliece, Rodemich, Ramsey, and Welch (MRRW) bound:*

$$\alpha_2(\delta) \leq h_2\big(\frac{1}{2} - \sqrt{\delta(1-\delta)}\big).$$

For the binary alphabet, these bounds are depicted in Figure A.1.

The Johnson Bound on List Decoding

Intuitively, it is natural to expect that a code with large minimum distance must remain a good list-decodable code when the list-decoding radius exceeds half the minimum distance. The Johnson bound makes this intuition rigorous. Below we quote a strengthened version of the bound.

Theorem A.1. *(cf.* [74, Section 3.3]*) Let $\mathcal{C}$ be a q-ary code of length n, and relative distance $\delta \geq (1-1/q)(1-\delta')$ for some $\delta' \in (0,1)$. Then for any $\gamma > \sqrt{\delta'}$, $\mathcal{C}$ is $((1-1/q)(1-\gamma), \ell)$ list-decodable for*

$$\ell = \min\{n(q-1), \frac{1-\delta'}{\gamma^2 - \delta'}\}.$$

Moreover, the code $\mathcal{C}$ is $((1-1/q)(1-\sqrt{\delta'}), 2n(q-1)-1)$ list-decodable. □

As an immediate corollary, we get that any binary code with relative distance at least $\frac{1}{2} - \epsilon$ is $(\frac{1}{2} - \sqrt{\epsilon}, \frac{1}{2\epsilon})$ list-decodable.

A.3 Reed-Solomon codes

Let $p = (p_1, \ldots, p_n)$ be a vector consisting of n distinct elements of $\mathbb{F}_q$ (assuming $q \geq n$). The *evaluation vector* of a polynomial $f \colon \mathbb{F}_q \to \mathbb{F}_q$ with respect to p is the vector $f(p) := (f(p_1), \ldots, f(p_n)) \in \mathbb{F}_q^n$.

A *Reed-Solomon* code of length n and dimension k over $\mathbb{F}_q$ is the set of evaluation vectors of all polynomials of degree at most $k-1$ over $\mathbb{F}_q$ with respect to a particular choice of p. The dimension of this code is equal to k. A direct corollary of Euclidean division algorithm states that, over any field, the number of zeros of any nonzero polynomial is less than or equal to its degree. Thus, we get that the minimum distance of a Reed-Solomon code is at least $n-k+1$, and because of the Singleton bound, is in fact equal to $n-k+1$. Hence we see that a Reed-Solomon code is MDS. A generator matrix for a Reed-Solomon code is given by the Vandermonde matrix

$$G := \begin{pmatrix} 1 & 1 & \ldots & 1 \\ p_1 & p_2 & \ldots & p_n \\ p_1^2 & p_2^2 & \ldots & p_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_1^{k-1} & p_2^{k-1} & \ldots & p_n^{k-1} \end{pmatrix}.$$

A.4 The Hadamard Code

The Hadamard code of dimension n is a linear binary code of length 2^n whose generator matrix can be obtained by arranging all binary sequences of length n as its columns. Each codeword of the Hadamard code can thus be seen as the truth table of a linear form

$$\ell(x_1, \ldots, x_n) = \sum_{i=1}^{n} \alpha_i x_i$$

over the binary field. Therefore, each nonzero codeword must have weight exactly 2^{n-1}, implying that the relative distance of the Hadamard code is $\frac{1}{2}$.

A.5 Concatenated Codes

Concatenation is a classical operation on codes that is mainly used for reducing the alphabet size of a code. Suppose that $\mathcal{C}_1$ (called the *outer code*) is an $(n_1, k_1, d_1)_Q$-code and $\mathcal{C}_2$ (called the *inner code*) is a $(n_2, k_2, d_2)_q$-code, where $Q = q^{k_2}$. The concatenation of $\mathcal{C}_1$ and $\mathcal{C}_2$, that we denote by $\mathcal{C}_1 \diamond \mathcal{C}_2$ is an $(n, k, d)_q$-code that can be conveniently defined by its encoder mapping as follows.

Let $x = (x_1, \ldots, x_{k_1}) \in [Q]^{k_1}$ be the message given to the encoder, and $C(x) = (c_1, \ldots, c_{n_1}) \in \mathcal{C}_1$ be its encoding under $\mathcal{C}_1$. Each c_i is thus an element

of $[q^{k_2}]$ and can thus be seen as a q-ary string of length k_2. Denote by $c'_i \in [q]^{n_2}$ be the encoding of this string under $\mathcal{C}_2$. Then the encoding of x by the concatenated code $\mathcal{C}_1 \diamond \mathcal{C}_2$ is the q-ary string of length $n_1 n_2$

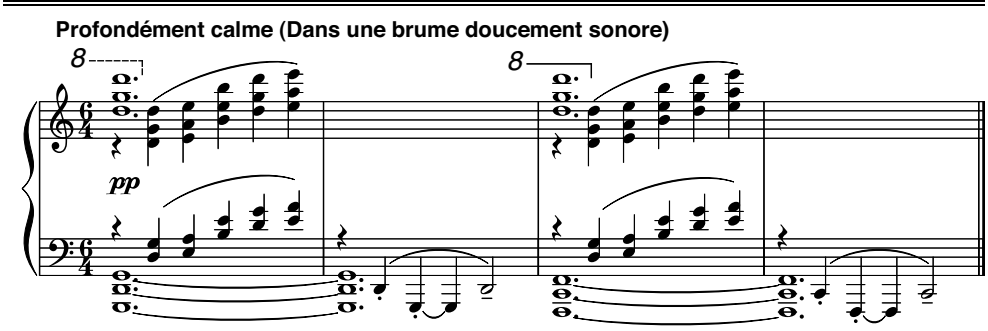
$$(c''_1, \ldots, c''_{n_1})$$

consisting of the string concatenation of symbol-wise encodings of $C(x)$ using $\mathcal{C}_2$.

Immediately from the above definition, one can see that $n = n_1 n_2$, and $k = k_1 k_2$. Moreover, it is straightforward to observe that the minimum distance of the concatenated code satisfies $d \geq d_1 d_2$. When $\mathcal{C}_1$ and $\mathcal{C}_2$ are linear codes, the so is $\mathcal{C}_1 \diamond \mathcal{C}_2$.

As an example, let $\mathcal{C}_1$ be a Reed-Solomon code of length $n_1 := 2^{k_2}$ and dimension $k_1 := 2\delta n_1$ over $\mathbb{F}_Q$, where $Q := 2^{k_2}$. Thus the relative distance of $\mathcal{C}_1$ equals $1 - 2\delta$. As the inner code $\mathcal{C}_2$, take the Hadamard code of dimension k_2 and length Q. The concatenated code $\mathcal{C} := \mathcal{C}_1 \diamond \mathcal{C}_2$ will thus have length $n := Qn_1 = 2^{2k_2}$, dimension $k := \delta k_2 2^{k_2+1}$, and relative distance at least $\frac{1}{2} - \delta$. Therefore, we obtain a binary $[n, k, d]$ code where $d \geq (\frac{1}{2} - \delta)n$, and $n \leq (k/\delta)^2$. By the Johnson bound (Theorem A.1), this code must be $(\frac{1}{2} - \delta, \ell)$ list-decodable with list size at most $1/(2\delta)$.

We remark that binary codes with relative minimum distance $\frac{1}{2} - \delta$ and rate $\Omega(\delta^3 \log(1/\delta))$ (which only depends on the parameter δ) can be obtained by concatenating Geometric Goppa codes on the Tsfasman-Vlăduţ-Zink bound (see Section 4.3.2) with the Hadamard code. The Gilbert-Varshamov bound implies that binary codes with relative distance $\frac{1}{2} - \delta$ and rate $\Omega(\delta^2)$ exist, and on the other hand, we know by the MRRW bound that $O(\delta^2 \log(1/\delta))$ is the best rate one can hope for.



Claude Debussy (1862–1918): Preludes, Book I, No. X (La cathédrale engloutie).

Bibliography

Number(s) following arrows (↑) indicate the page number(s) where the citation appears.

[1] M. Adler, E.D. Demaine, N.J.A. Harvey, and M. Pătraşcu, *Lower bounds for asymmetric communication channels and distributed source coding*, Proceedings of the 17th symposium on discrete algorithms (SODA), 2006, pp. 251–260. ↑24

[2] M. Adler and B. Maggs, *Protocols for asymmetric communication channels*, Journal of Computer and System Sciences **63** (2001), no. 4, 573––596. ↑24

[3] M. Agrawal, N. Kayal, and N. Saxena, *PRIMES is in P*, Annals of Mathematics **160** (2004), no. 2, 781–793. ↑2, 10

[4] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, *Network information flow*, IEEE Transactions on Information Theory **46** (2000), no. 4, 1204–1216. ↑64

[5] N. Alon and J.H. Spencer, *The probabilistic method*, John Wiley and Sons, 2000. ↑3, 11

[6] E. Ardestanizadeh, M. Cheraghchi, and A. Shokrollahi, *Bit precision analysis for compressed sensing*, Proceedings of IEEE international symposium on information theory (ISIT), 2009. ↑v

[7] E. Arıkan, *Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels*, IEEE Transactions on Information Theory **55** (2009), no. 7, 3051–3073. ↑127

[8] S. Arora and B. Barak, *Computational complexity: A modern approach*, Cambridge University Press, 2009. ↑11

[9] B. Barak, A. Rao, R. Shaltiel, and A. Wigderson, *2-source dispersers for sub-polynomial entropy and Ramsey graphs beating the Frankl-Wilson construction*, Proceedings of the 38th annual ACM symposium on theory of computing (STOC), 2006, pp. 671–680. ↑4, 6

[10] A. Ben-Aroya and A. Ta-Shma, *Constructing small-bias sets from algebraic-geometric codes*, Proceedings of the 50th annual IEEE symposium on foundations of computer science (FOCS), 2009. ↑109

[11] C.H. Bennett, G. Brassard, and J-M. Robert, *How to reduce your enemy's information*, Proceedings of the 5th annual international cryptology conference (CRYPTO), 1985, pp. 468–476. ↑44

[12] E.R. Berlekamp, R. McEliece, and H.C.A van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Transactions on Information Theory **24** (1978), 384–386. ↑172

[13] R.E. Blahut, *Theory and practice of error control codes*, Addison-Wesley, 1983. ↑126

[14] A. Blass and Y. Gurevich, *Pairwise testing*, Bulletin of the EATCS **78** (2002), 100–132. ↑77

[15] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudorandom bits*, SIAM Journal on Computing **13** (1984), no. 4, 850–864. ↑154

[16] J. Bourgain, *On the construction of affine extractors*, Geometric and Functional Analysis **17** (2007), no. 1, 33–57. ↑49, 62

[17] ———, *Personal Communication*, 2008. ↑62

[18] W.J. Bruno, E. Knill, D.J. Balding, D.C. Bruce, N.A. Doggett, W.W. Sawhill, R.L. Stallings, C.C. Whittaker, and D.C. Torney, *Efficient pooling designs for library screening*, Genomics **26** (1995), no. 1, 21–30. ↑77

[19] H. Buhrman, P.B. Miltersen, J. Radhakrishnan, and S. Venkatesh, *Are bitvectors optimal?*, SIAM Journal on Computing **31** (2002), no. 6, 1723–1744. ↑102

[20] J. Buresh-Oppenheim, V. Kabanets, and R. Santhanam, *Uniform hardness amplification in* NP *via monotone codes*, 2006. *ECCC* Technical Report TR06-154. ↑88

[21] N. Cai and R. W. Yeung, *Secure network coding*, Proceedings of IEEE international symposium on information theory (ISIT), 2002. ↑65

[22] R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai, *Exposure-resilient functions and all-or-nothing transforms*, Proceedings of the 19th annual international cryptology conference (CRYPTO), 1999, pp. 503–518. ↑44, 46, 47, 68

[23] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, *Randomness conductors and constant-degree expansion beyond the degree/2 barrier*, Proceedings of the 34th annual ACM symposium on theory of computing (STOC), 2002, pp. 659–668. ↑25, 98, 99, 120

[24] H. Chang, H-B. Chen, H-L. Fu, and C-H. Shi, *Reconstruction of hidden graphs and threshold group testing*, Journal of Combinatorial Optimization (2010). ↑103

[25] D. X. Charles, K. E. Lauter, and E. Z. Goren, *Cryptographic hash functions from expander graphs*, Journal of Cryptology (2007). ↑56

[26] H-B. Chen, D-Z. Du, and F-K. Hwang, *An unexpected meeting of four seemingly unrelated problems: graph testing, DNA complex screening, superimposed codes and secure key distribution*, Journal of Combinatorial Optimization **14** (2007), no. 2-3, 121–129. ↑77, 104, 106, 123

[27] H-B. Chen and H-L. Fu, *Nonadaptive algorithms for threshold group testing*, Discrete Applied Mathematics **157** (2009), 1581–1585. ↑103, 105, 106, 167

[28] H-B. Chen, H-L. Fu, and F-K. Hwang, *An upper bound of the number of tests in pooling designs for the error-tolerant complex model*, Optimization Letters **2** (2008), no. 3, 425–431. ↑106, 123

[29] Y. Cheng and D-Z. Du, *New constructions of one- and two-stage pooling designs*, Journal of Computational Biology **15** (2008), no. 2, 195–205. ↑115

[30] M. Cheraghchi, *Noise-resilient group testing: Limitations and constructions*, Proceedings of the 17th international symposium on fundamentals of computation theory (FCT), 2009, pp. 62–73. ↑vi

[31] ———, *Improved constructions for non-adaptive threshold group testing*, Proceedings of the 37th international colloquium on automata, languages and programming (ICALP), 2010. arXiv: cs.DM/1002.2244. ↑vi

[32] M. Cheraghchi, F. Didier, and A. Shokrollahi, *Invertible extractors and wiretap protocols*, Proceedings of IEEE international symposium on information theory (ISIT), 2009. ↑iv

[33] M. Cheraghchi, A. Hormati A. Karbasi, and M. Vetterli, *Compressed sensing with probabilistic measurements: A group testing solution*, Proceedings of the annual Allerton conference on communication, control, and computing, 2009. ↑v

[34] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama, *Graph-constrained group testing*, Proceedings of IEEE international symposium on information theory (ISIT), 2010. ↑v

[35] B. Chor and O. Goldreich, *Unbiased bits from sources of weak randomness and probabilistic communication complexity*, SIAM Journal of Computing **17** (1988), no. 2, 230–261. ↑4, 6

[36] B. Chor, O. Goldreich, J. Håstad, J. Friedmann, S. Rudich, and R. Smolensky, *The bit extraction problem or t-resilient functions*, Proceedings of the 26th annual IEEE symposium on foundations of computer science (FOCS), 1985, pp. 396–407. ↑44, 45

[37] R. Clifford, K. Efremenko, E. Porat, and A. Rothschild, *k-mismatch with don't cares*, Proceedings of the 15th european symposium on algorithm (ESA), 2007, pp. 151–162. ↑77

[38] G. Cormode and S. Muthukrishnan, *What's hot and what's not: tracking most frequent items dynamically*, ACM Transactions on Database Systems **30** (2005), no. 1, 249–278. ↑77

[39] ———, *Combinatorial algorithms for compressed sensing*, Proceedings of information sciences and systems, 2006, pp. 198–201. ↑77

[40] T.M. Cover and J.A. Thomas, *Elements of information theory*, Second, John Wiley and Sons, 2006. ↑17, 24, 131, 132

[41] I. Csiszár and J. Körner, *Broadcast channels with confidential messages*, IEEE Transactions on Information Theory **24** (1978), no. 3, 339–348. ↑64

[42] P. Damaschke, *Threshold group testing*, General theory of information transfer and combinatorics, 2006, pp. 707–718. ↑103

[43] A. De Bonis, L. Gasieniec, and U. Vaccaro, *Optimal two-stage algorithms for group testing problems*, SIAM Journal on Computing **34** (2005), no. 5, 1253–1270. ↑123

[44] M. DeVos and A. Gabizon, *Simple affine extractors using dimension expansion*, Proceedings of the 25th IEEE conference on computational complexity (CCC), 2010. ↑49

[45] Y. Dodis, *Exposure-resilient cryptography*, Ph.D. Thesis, 2000. ↑7, 44

[46] ———, *On extractors, error-correction and hiding all partial information*, Proceedings of the IEEE information theory workshop (ITW), 2005. ↑53

[47] Y. Dodis, A. Sahai, and A. Smith, *On perfect and adaptive security in exposure-resilient cryptography*, Proceedings of Eurocrypt, 2001, pp. 301–324. ↑43, 45, 54

[48] Y. Dodis and A. Smith, *Entropic security and the encryption of high-entropy messages*, Proceedings of the theory of cryptography conference (TCC), 2005, pp. 556–577. ↑53

[49] R. Dorfman, *The detection of defective members of large populations*, Annals of Mathematical Statistics **14** (1943), 436–440. ↑76

[50] D-Z. Du and F. Hwang, *Combinatorial group testing and its applications*, Second, World Scientific, 2000. ↑11, 77, 104, 123

[51] D-Z. Du and F-K. Hwang, *Pooling designs and nonadaptive group testing*, World Scientific, 2006. ↑11, 77, 104

[52] I. Dumer, D. Micciancio, and M. Sudan, *Hardness of approximating the minimum distance of a linear code*, IEEE Transactions on Information Theory **49** (2003), no. 1, 22–37. ↑161

[53] A. D'yachkov, P. Vilenkin, A. Macula, and D. Torney, *Families of finite sets in which no intersection of ℓ sets is covered by the union of s others*, Journal of Combinatorial Theory, Series A **99** (2002), 195–218. ↑106, 123

[54] A.G. D'yachkov and V.V. Rykov, *Bounds of the length of disjunct codes*, Problems of Control and Information Theory **11** (1982), 7–13. ↑83, 102

[55] S. Y. El Rouayheb and E. Soljanin, *On wiretap networks II*, Proceedings of IEEE international symposium on information theory (ISIT), 2007, pp. 24–29. ↑65

[56] D. Eppstein, M.T. Goodrich, and D.S. Hirschberg, *Improved combinatorial group testing algorithms for real-world problem sizes*, SIAM Journal on Computing **36** (2007), no. 5, 1360–1375. ↑123

[57] P. Erdős, *Some remarks on the theory of graphs*, Bulletin of the American Mathematical Society **53** (1947), 292–294. ↑4

[58] M. Farach, S. Kannan, E. Knill, and S. Muthukrishnan, *Group testing problems with sequences in experimental molecular biology*, Proceedings of compression and complexity of sequences, 1997, pp. 357–367. ↑77

[59] J. Feldman, T. Malkin, R. Servedio, and C. Stein, *On the capacity of secure network coding*, Proceedings of the annual Allerton conference on communication, control, and computing, 2004. ↑65

[60] G.D. Forney, *Concatenated codes*, MIT Press, 1966. ↑127, 142, 143

[61] K. Friedl and S-C. Tsai, *Two results on the bit extraction problem*, Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science **99** (2000). ↑44

[62] J. Friedmann, *On the bit extraction problem*, Proceedings of the 33rd annual IEEE symposium on foundations of computer science (FOCS), 1992, pp. 314–319. ↑44

[63] Z. Füredi, *On r-cover-free families*, Journal of Combinatorial Theory, Series A **73** (1996), 172–173. ↑83

[64] E.M. Gabidulin, *Theory of codes with maximum rank distance (translation)*, Problems of Information Transmission **21** (1985), no. 1, 1–12. ↑51

[65] A. Gabizon and R. Raz, *Deterministic extractors for affine sources over large fields*, Proceedings of the 46th annual IEEE symposium on foundations of computer science (FOCS), 2005, pp. 407–418. ↑49, 59, 60, 62

[66] H. Gao, F.K. Hwang, M.T. Thai, W. Wu, and T. Znati, *Construction of $d(H)$-disjunct matrix for group testing in hypergraphs*, Journal of Combinatorial Optimization **12** (2006), 297–301. ↑104, 106

[67] A. Garcia and H. Stichtenoth, *A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlăduţ bound*, Invent. Math. **121** (1995), 211–222. ↑108, 151

[68] E.N. Gilbert, *A comparison of signaling alphabets*, Bell System Technical Journal **31** (1952), 504–522. ↑47, 150, 159

[69] M.X. Goemans and D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, Journal of the ACM **42** (1995), 1115–1145. ↑2

[70] O. Goldreich, *Computational complexity: A conceptual perspective*, Cambridge University Press, 2008. ↑9, 11

[71] O. Goldreich and D. Zuckerman, *Another proof that* BPP $\subseteq$ PH *(and more)*, 1997. *ECCC* Technical Report TR97-045 (available online at `http://eccc.hpi-web.de/eccc-reports/1997/TR97-045/index.html`). ↑7

[72] V.D. Goppa, *Codes on algebraic curves*, Soviet Mathematics Doklady **24** (1981), 170–172. ↑47, 151

[73] R. Gradwohl, S. Vadhan, and D. Zuckerman, *Random selection with an adversarial majority*, Proceedings of the 26th annual international cryptology conference (CRYPTO), 2006. ↑7

[74] V. Guruswami, *List decoding of error-correcting codes*, Ph.D. Thesis, 2001. ↑11, 101, 174

[75] V. Guruswami and P. Gopalan, *Hardness amplification within* NP *against deterministic algorithms*, Proceedings of the 23rd IEEE conference on computational complexity (CCC), 2008, pp. 19–30. ↑88

[76] V. Guruswami and P. Indyk, *Linear-time encodable/decodable codes with near-optimal rate*, IEEE Transactions on Information Theory **51** (2005), no. 10, 3393–3400. ↑142

[77] V. Guruswami and A. Rudra, *Concatenated codes can achieve list decoding capacity*, Proceedings of the 19th symposium on discrete algorithms (SODA), 2008. ↑101

[78] V. Guruswami, C. Umans, and S. Vadhan, *Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes*, Journal of the ACM **56** (2009), no. 4. ↑35, 37, 38, 39, 91, 93, 101, 167

[79] J. Håstad, *Some optimal inapproximability results*, Journal of the ACM **48** (2001), no. 4, 798–859. Preliminary version in *Proceedings of STOC'97*. ↑2

[80] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby, *A pseudorandom generator from any one-way function*, SIAM Journal on Computing **28** (1999), no. 4, 1364–1396. ↑154

[81] E-S. Hong and R.E. Ladner, *Group testing for image compression*, Data compression conference, 2000, pp. 3–12. ↑77

[82] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bulletin of the American Mathematical Society **43** (2006), no. 4, 439–561. ↑11, 55

[83] J. Hopcroft, W. Paul, and L. Valiant, *On time versus space*, Journal of the ACM **24** (1977), 332–337. ↑157

[84] R. Impagliazzo, L. Levin, and M. Luby, *Pseudorandom generation from one-way functions*, Proceedings of the 21st annual ACM symposium on theory of computing (STOC), 1989, pp. 12–24. ↑26

[85] R. Impagliazzo and A. Wigderson, $\mathsf{P} = \mathsf{BPP}$ *unless* E *has sub-exponential circuits: derandomizing the XOR lemma*, Proceedings of the 29th annual ACM symposium on theory of computing (STOC), 1997, pp. 220–229. ↑157, 161

[86] P. Indyk, *Explicit construction of compressed sensing of sparse signals*, Proceedings of the 19th symposium on discrete algorithms (SODA), 2008. ↑7

[87] J. Justesen, *A class of constructive asymptotically good algebraic codes*, IEEE Transactions on Information Theory **18** (1972), 652–656. ↑127, 142

[88] J. Kamp and D. Zuckerman, *Deterministic extractors for bit-fixing sources and exposure-resilient cryptography*, SIAM Journal on Computing **36** (2006), 1231–1247. ↑55, 56

[89] W.H. Kautz and R.C. Singleton, *Nonrandom binary superimposed codes*, IEEE Transactions on Information Theory **10** (1964), 363–377. ↑100, 102, 106, 107, 108

[90] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, *Optimal inapproximability results for MAX-CUT and other two-variable CSPs?*, Proceedings of the 45th annual IEEE symposium on foundations of computer science (FOCS), 2004, pp. 146–154. ↑2

[91] H.K. Kim and V. Lebedev, *On optimal superimposed codes*, Journal of Combinatorial Designs **12** (2004), 79–91. ↑106, 123

[92] A. Klivans and D. van Melkebeek, *Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses*, SIAM Journal on Computing **31** (2002), no. 5, 1501–1526. ↑161

[93] E. Knill, *Lower bounds for identifying subset members with subset queries*, Proceedings of the 6th symposium on discrete algorithms (SODA), 1995, pp. 369–377. ↑88

[94] R. Kötter and M. Médard, *An algebraic approach to network coding*, IEEE/ACM Transactions on Networking **11** (2003), no. 5, 782–795. ↑64

[95] F. R. Kschischang and D. Silva, *Security for wiretap networks via rank-metric codes*, 2007. Unpublished manuscript (arXiv: cs.IT/0801.0061). ↑66, 67

[96] K. Kurosawa, T. Johansson, and D. Stinson, *Almost k-wise independent sample spaces and their cryptologic applications*, Journal of Cryptology **14** (2001), no. 4, 231–253. ↑44, 47

[97] S-Y. R. Li, R. W. Yeung, and N. Cai, *Linear network coding*, IEEE Transactions on Information Theory **49** (2003), no. 2, 371–381. ↑64

[98] J. H. van Lint, *Introduction to coding theory*, Third, Graduate Texts in Mathematics, vol. 86, Springer Verlag, 1998. ↑11, 170

[99] L. Lovász, *Random walks on graphs: A survey*, Combinatorics, Paul Erdős is Eighty, Vol. 2 (ed. D. Miklós, V. T. Sós, T. Szőnyi), János Bolyai Mathematical Society, Budapest (1996), 353–398. ↑56

[100] A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica **8** (1988), 261–277. ↑59

[101] M. Luby, *LT-codes*, Proceedings of the 43rd annual IEEE symposium on foundations of computer science (FOCS), 2002, pp. 271–280. ↑63

[102] A.J. Macula, *Probabilistic nonadaptive group testing in the presence of errors and DNA library screening*, Annals of Combinatorics **3** (1999), no. 1, 61–69. ↑77

[103] F.J. MacWilliams and N.J. Sloane, *The theory of error-correcting codes*, North Holand, 1977. ↑11, 110, 170

[104] S. Mahajan and H. Ramesh, *Derandomizing semidefinite programming based approximation algorithms*, Proceedings of the 36th annual IEEE symposium on foundations of computer science (FOCS), 1995, pp. 162–169. ↑2

[105] Yu.I. Manin, *What is the maximum number of points on a curve over* $\mathbb{F}_2$*?*, Journal of the Faculty of Science, University of Tokyo **28** (1981), 715–720. ↑150

[106] J.L. Massey, *Threshold decoding*, MIT Press, Cambridge, Massachusetts, USA, 1963. ↑160

[107] G.L. Miller, *Riemann's hypothesis and tests for primality*, Journal of Computer and System Sciences **13** (1976), no. 3, 300–317. ↑2

[108] P.B. Miltersen, *Derandomizing complexity classes*, Book chapter in *Handbook of Randomized Computing*, Kluwer Academic Publishers (2001). ↑158

[109] M. Mitzenmacher and E. Upfal, *Probability and computing*, Cambridge University Press, 2005. ↑11, 55

[110] M. Morgenstern, *Existence and explicit constructions of* $q+1$ *regular ramanujan graphs for every prime power* q, Journal of Combinatorial Theory, Series B **62** (1994), 44–62. ↑59

[111] E. Mossel and C. Umans, *On the complexity of approximating the VC dimension*, Proceedings of the 16th IEEE conference on computational complexity (CCC), 2001, pp. 220–225. ↑7

[112] R. Motwani and P. Raghavan, *Randomized algorithms*, Cambridge University Press, 1995. ↑11, 55

[113] H-Q. Ngo and D-Z. Du, *A survey on combinatorial group testing algorithms with applications to DNA library screening*, DIMACS Series on Discrete Math. and Theoretical Computer Science **55** (2000), 171–182. ↑77

[114] A. Nilli, *On the second eigenvalue of a graph*, Discrete Mathematics **91** (1991), 207–210. ↑58

[115] N. Nisan and A. Wigderson, *Hardness vs. randomness*, Journal of Computer and Systems Sciences **49** (1994), no. 2, 149–167. ↑10, 151, 154, 155, 161

[116] L. H. Ozarow and A. D. Wyner, *Wire-tap channel II*, AT&T Bell Laboratories Technical Journal **63** (1984), 2135–2157. ↑41, 46, 55, 65

[117] C.H. Papadimitriou, *Computational complexity*, Addison-Wesley, 1994. ↑152

[118] F. Parvaresh and A. Vardy, *Correcting errors beyond the Guruswami-Sudan radius in polynomial time*, Proceedings of the 46th annual IEEE symposium on foundations of computer science (FOCS), 2005, pp. 285–294. ↑35, 100

[119] A. K. Pizer, *Ramanujan graphs and Hecke operators*, Bulletin of the American Mathematical Society **23** (1990), no. 1, 127–137. ↑59

[120] E. Porat and A. Rothschild, *Explicit non-adaptive combinatorial group testing schemes*, Proceedings of the 35th international colloquium on automata, languages and programming (ICALP), 2008, pp. 748–759. ↑106, 110, 122

[121] M. Rabin, *Probabilistic algorithm for testing primality*, Journal of Number Theory **12** (1980), no. 1, 128–138. ↑2

[122] J. Radhakrishan and A. Ta-Shma, *Tight bounds for depth-two superconcentrators*, Proceedings of the 38th annual IEEE symposium on foundations of computer science (FOCS), 1997, pp. 585–594. ↑25, 120

[123] R. Raz, O. Reingold, and S. Vadhan, *Extracting all the randomness and reducing the error in Trevisan's extractor*, Journal of Computer and System Sciences **65** (2002), no. 1, 97–128. ↑34, 60, 99

[124] R. Raz and A. Yehudayoff, *Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors*, Proceedings of the 49th annual IEEE symposium on foundations of computer science (FOCS), 2008. ↑6

[125] T. Richardson and R. Urbanke, *Modern coding theory*, Cambridge University Press, 2008. ↑126

[126] R. Rivest, *All-or-nothing encryption and the package transform*, Proceedings of the international workshop on fast software encryption, 1997, pp. 210–218. ↑44, 45

[127] R.M. Roth, *Introduction to coding theory*, Cambridge University Press, 2006. ↑11, 127, 170

[128] A. Russell and D. Zuckerman, *Perfect information leader election in* $\log^* n + o(1)$ *rounds*, Journal of Computer and Systems Sciences **63** (2001), 612–626. ↑7

[129] Ruszinkó, *On the upper bound of the size of the* r*-cover-free families*, Journal of Combinatorial Theory, Series A **66** (1994), 302–310. ↑83

[130] A. Schliep, D. Torney, and S. Rahmann, *Group testing with DNA chips: Generating designs and decoding experiments*, Proceedings of computational systems bioinformatics, 2003. ↑77

[131] J. T. Schwartz, *Fast probabilistic algorithms for verification of polynomial identities*, Journal of the ACM **27** (1980), no. 4, 701–717. ↑3

[132] R. Shaltiel, *How to get more mileage from randomness extractors*, Proceedings of the 21st annual conference on computational complexity, 2006, pp. 46–60. ↑59, 60, 165

[133] ———, *Weak derandomization of weak algorithms: explicit versions of yao's lemma*, Proceedings of the 24th annual conference on computational complexity, 2009. ↑165

[134] R. Shaltiel and C. Umans, *Simple extractors for all min-entropies and a new pseudorandom generator*, Journal of the ACM **52** (2005), no. 2, 172–216. ↑60, 99

[135] A. Shamir, *On the generation of cryptographically strong pseudorandom sequences*, ACM Transactions on Computer Systems **1** (1983), no. 1, 38–44. ↑154

[136] C.E. Shannon, *A mathematical theory of communication*, The Bell System Technical Journal **27** (1948), 379–423 and 623–656. ↑5, 126, 131

[137] A. Shokrollahi, *Raptor codes*, IEEE Transactions on Information Theory **52** (2006), 2551–2567. ↑63, 126

[138] V. Shoup, *New algorithms for finding irreducible polynomials over finite fields*, Mathematics of Computation **54** (1990), 435–447. ↑38

[139] M. Sipser, *Introduction to the theory of computation*, Second, Course Technology, 2005. ↑152

[140] R.M. Solovay and V. Strassen, *A fast Monte-Carlo test for primality*, SIAM Journal on Computing **6** (1977), no. 1, 84–85. ↑2

[141] D. Spielman, *Linear-time encodable and decodable error-correcting codes*, IEEE Transactions on Information Theory **42** (1996), 1723–1731. ↑142

[142] H. Stichtenoth, *Algebraic function fields and codes*, Springer, 1993. ↑109

[143] D. Stinson, *Resilient functions and large set of orthogonal arrays*, Congressus Numerantium **92** (1993), 105–110. ↑44

[144] D.R. Stinson and R. Wei, *Generalized cover-free families*, Discrete Mathematics **279** (2004), 463–477. ↑106, 123

[145] D.R. Stinson, R. Wei, and L. Zhu, *Some new bounds for cover-free families*, Journal of Combinatorial Theory, Series A **90** (2000), 224–234. ↑106, 123

[146] M. Sudan, L. Trevisan, and S. Vadhan, *Pseudorandom generators without the XOR lemma*, Journal of Computer and Systems Sciences **62** (2001), no. 2, 236–266. ↑157

[147] A. Ta-Shma, *Storing information with extractors*, Information Processing Letters **83** (2002), no. 5, 267–274. ↑7, 102

[148] A. Ta-Shma, C. Umans, and D. Zuckerman, *Lossless condensers, unbalanced expanders, and extractors*, Proceedings of the 33rd annual ACM symposium on theory of computing (STOC), 2001, pp. 143–152. ↑100

[149] A. Ta-Shma and D. Zuckerman, *Extractor codes*, IEEE Transactions on Information Theory **50** (2004), no. 12, 3015–3025. ↑7, 91, 93, 99, 101

[150] A. Ta-Shma, D. Zuckerman, and S. Safra, *Extractors from Reed-Muller codes*, Journal of Computer and System Sciences **72** (2006), 786–812. ↑60

[151] J-P. Tillich and G. Zémor, *Collisions for the lps expander graph hash function*, Proceedings of Eurocrypt, 2008, pp. 254–269. ↑56

[152] L. Trevisan, *Extractors and pseudorandom generators*, Journal of the ACM **48** (2001), no. 4, 860–879. ↑28, 32, 34, 60, 99

[153] L. Trevisan and S. Vadhan, *Extracting randomness from samplable distributions*, Proceedings of the 41st annual IEEE symposium on foundations of computer science (FOCS), 2000, pp. 32–42. ↑19, 165

[154] M.A. Tsfasman, S.G. Vlăduţ, and Th. Zink, *Modular curves, Shimura curves, and Goppa codes better than the Varshamov-Gilbert bound*, Mathematische Nachrichten **109** (1982), 21–28. ↑47, 108, 151, 168

[155] S. Vadhan, *The unified theory of pseudorandomness*, Proceedings of the international congress of mathematicians, 2010. ↑91

[156] A. Vardy, *The intractability of computing the minimum distance of a code*, IEEE Transactions on Information Theory **43** (1997), no. 6, 1757–1766. ↑161

[157] R. R. Varshamov, *Estimate of the number of signals in error correcting codes*, Doklady Akademii Nauk SSSR **117** (1957), 739–741. ↑47, 150, 159, 160

[158] U.V. Vazirani, *Towards a strong communication complexity theory or generating quasi-random sequences from two communicating semi-random sources*, Combinatorica **7** (1987), no. 4, 375–392. ↑44

[159] J. von Neumann, *Various techniques used in connection with random digits*, Applied Math Series **12** (1951), 36–38. ↑12

[160] J. Watkinson, M. Adler, and F.E. Fich, *New protocols for asymmetric communication channels*, Proceedings of international colloquium on structural information and communication complexity (SIROCCO), 2001. ↑24

[161] A. Wigderson and D. Zuckerman, *Expanders that beat the eigenvalue bound: explicit construction and applications*, Combinatorica **19** (1999), no. 1, 125–138. ↑6

[162] J. Wolf, *Born-again group testing: multiaccess communications*, IEEE Transactions on Information Theory **31** (1985), 185–191. ↑77

[163] W. Wu, Y. Huang, X. Huang, and Y. Li, *On error-tolerant DNA screening*, Discrete Applied Mathematics **154** (2006), no. 12, 1753–1758. ↑77

[164] W. Wu, Y. Li, C.H. Huang, and D.Z. Du, *Molecular biology and pooling design*, Data Mining in Biomedicine **7** (2008), 133–139. ↑77

[165] A. D. Wyner, *The wire-tap channel*, The Bell System Technical Journal **54** (1975), 1355–1387. ↑40

[166] A. Yao, *Theory and applications of trapdoor functions*, Proceedings of the 23rd annual IEEE symposium on foundations of computer science (FOCS), 1982, pp. 80–91. ↑154

[167] A. Yehudayoff, *Affine extractors over prime fields*, 2009. Unpublished manuscript. ↑49, 62

[168] R. W. Yeung, S-Y. R. Li, N. Cai, and Z. Zhang, *Network coding theory*, Foundations and Trends in Communications and Information Theory, 2005. ↑64

[169] R. E. Zippel, *Probabilistic algorithms for sparse polynomials*, Springer Lecture Notes in Computer Science (EUROSCAM'79) **72** (1979), 216–226. ↑3

[170] D. Zuckerman, *On unapproximable versions of* NP*-complete problems*, SIAM Journal of Computing **25** (1996), 1293–1304. ↑7

[171] ———, *Randomness-optimal oblivious sampling*, Random Structures and Algorithms **11** (1997), no. 4, 345–367. ↑7

[172] ———, *Linear degree extractors and the inapproximability of Max Clique and Chromatic Number*, Theory of Computing **3** (2007), no. 6, 103–128. ↑56

Index