CAREER: A Unified Theory of Pseudorandomness

Salil P. Vadhan
Division of Engineering and Applied Sciences
Harvard University

Project Summary

Randomization is one of the most pervasive paradigms in computer science, with widespread use in areas including algorithm design, cryptography, coding theory, network design, and interactive proofs. However, it is still not known to what extent the randomness is really *necessary* in these settings, and understanding this is important for both practical and theoretical reasons. The main approach to addressing this issue is via the paradigm of *pseudorandomness*, which is the theory of generating objects that "look random" despite being constructed using little or no randomness.

The proposed research builds upon recently established connections between four of the most important kinds of pseudorandom objects: *pseudorandom generators* — procedures which stretch a short "seed" of truly random bits into a long string of "pseudorandom" bits which cannot be distinguished from truly random by any efficient algorithm, *expander graphs* — networks which are sparse but nevertheless highly connected, *error-correcting codes* — methods for encoding messages so that even if many of the symbols are corrupted, the original message can still be recovered, and *extractors* — procedures which extract almost uniformly distributed bits from a source of biased and correlated bits. While these objects were previously each the subject of largely distinct bodies of research, several recent discoveries have shown that these objects are almost *the same* when interpreted appropriately. This unification makes the time ripe for substantial advances in the area, and also for educating a larger community about the topic. With this in mind, the broad goals of this career development plan are the following:

- Understand more fully and strengthen the connections between these various pseudorandom objects.
- Exploit these connections to attack some of the important open problems in the area (such as the construction of constant-degree expander graphs with near-optimal expansion, the complete derandomization of space-bounded algorithms, and determining to what extent circuit lower bounds are necessary for derandomizing time-bounded algorithms).
- Make use of the improved understanding of pseudorandomness to improve and extend its applications in other areas, such as cryptography, complexity theory, and data structures.
- Involve students at all levels of the research, from testing potential constructions through computer experiments, as might be suitable for undergraduates, to the mathematical research suitable for Ph.D. students.
- Convey the unified theory of pseudorandomness that we are developing to a larger community. This will be done by developing a new graduate course on the pseudorandomness (with publicly available lecture notes) and filtering aspects of the theory into undergraduate and graduate courses on related topics (such as a new undergraduate course on cryptography).

1 Introduction

During the past few decades, *randomization* has become one of the most pervasive paradigms in computer science. Its widespread use includes:

Algorithm Design For the problem of testing whether a number is prime, which arises often in cryptography, the only efficient algorithms known require generating random numbers [SS77, Mil76, Rab80]. Similarly, the Markov Chain Monte Carlo Method is the only way we know to efficiently approximate for various quantities arising in statistical physics, such as the matrix permanent [JS89, JSV01].

Cryptography Randomness is woven into the very way in which we define security. Indeed, a “secret key” is not secret if it is fixed rather than random. Moreover, often the cryptographic algorithms themselves must be randomized to achieve satisfactory levels of security (cf., [GM84]).

Randomized Constructions Many useful combinatorial objects can be constructed simply by generating them at random. (This is known as the *Probabilistic Method* [AS00].) Examples include *error-correcting codes* for communicating on noisy channels [Sha48] and various kinds of fault-tolerant networks (known as *expander graphs*) [Pin73].

Interactive Proofs Randomization, together with interactive communication, can also be very useful when one party wishes to convince another of some assertion [GMR89, BM88]. Verifying such *interactive proofs* can often be much more efficient than verifying classical “written” proofs [LFKN92, Sha92]. Interactive proofs can also have properties (such as leaking “zero knowledge”) that make them widely applicable in cryptography [GMR89, GMW91].

So randomness *appears* to be extremely useful in these settings, but we still do not know to what extent it is really *necessary*. Thus, the proposed research is driven by the following question.

Motivating Question: *Can we reduce or even eliminate the need for randomness in the above settings?*

There are several reasons for pursuing this research direction. First, essentially all of the applications of randomness assume that we have a source of *perfect randomness* — one that gives “coin tosses” that are completely *unbiased* and *independent* of each other. It is unclear whether physical sources of perfect randomness exist and are inexpensive to access. The sources of randomness our computers can directly access in practice, such as statistics on mouse movements, clearly contain biases and dependencies. Second, randomized constructions of useful objects often do not provide us with efficient algorithms for using them; indeed, even writing down a description of a randomly selected object can be infeasible. For example, for two parties to use an error-correcting code to communicate, they need to share a short description of the code that allows them to efficiently perform encoding and decoding. Finally, randomization has become a central aspect of computation, so our understanding of computation would be incomplete without understanding the power randomness provides.

Over two decades, an exciting body of research has developed to address the Motivating Question via a powerful paradigm known as *pseudorandomness*. This is the theory of efficiently generating objects that “look random”, despite being constructed using little or no randomness. This body of work has yielded efficient constructions of various kinds of “pseudorandom” objects. Some of the most important of these are:

Pseudorandom Generators These are procedures which stretch a short “seed” of truly random bits into a long string of “pseudorandom” bits which cannot be distinguished from truly random by any efficient algorithm [BM84, Yao82, NW94]. Pseudorandom generators make it possible to automatically reduce, and often completely eliminate, the randomness used by any efficient algorithm [Yao82]. Moreover, pseudorandom generators are a fundamental tool in cryptography, and indeed this was the original motivation for their definition. Many of the problems of private-key cryptography can be viewed as generating a lot of unpredictable material (for encryption, signing, etc.) from a short, truly random shared key.

Expander Graphs Expanders are graphs which are sparse but nevertheless highly connected. Such graphs have been used to address many fundamental problems in computer science, on topics including network design (e.g. [Pip87, PY82, AKS83]), complexity theory ([Val77, Sip88, Urq87]), derandomization ([NN93, INW94, IW97]), coding theory ([Tan81, SS96, Spi96]), cryptography ([GIL$^+$90]), and computational group theory ([LP01]). They also have recently been shown to have unexpected connections with deep questions in pure mathematics ([Gro00]).

Error-Correcting Codes These are methods for encoding messages so that even if many of the symbols are corrupted, the original message can still be recovered. Originally motivated by the problem of communicating over a noisy channel [Sha48], they have turned out to have a wide variety of other applications in computer science. Here we focus on "the highly noisy case," where there are so many corruptions that uniquely decoding the original message is impossible, but it is still possible to produce a short list of possible candidates. Efficient algorithms for this *list decoding* problem have only been developed fairly recently [Sud97, GS99], but still they have already found many applications (cf., [Sud00]).

Extractors These are procedures which extract almost uniformly distributed bits from a source of biased and correlated bits, using a small number of truly random bits as a catalyst [NZ96]. Their original motivation was to allow us to use randomized algorithms even with imperfect physical sources of randomness [Zuc96]. In addition to this direct application, they have turned out to have a wide variety of other applications in computer science, such as leader election [Zuc97, RZ98]; hardness of approximation [Zuc96, Uma99]; cryptography [CDH$^+$00, MW00]; pseudorandom generators [NZ96, ACR97, RR99, STV01]; and other problems in complexity theory [Sip88, GZ97].

We stress that while we have described all of these objects qualitatively, formally they are each defined by several parameters and the trade-offs between these parameters are of vital importance in their applications. For sake of readability, we will avoid getting into the details of these parameters except when absolutely necessary. We will occasionally refer to the goal of obtaining an 'optimal' construction, which in most cases is an important open problem.

Due to their fundamental nature and wide applicability, each of the above objects has been the center of a large body of research. Until recently these four bodies of research were largely distinct. While it was common to use one of them as a *tool* to construct another (e.g., expander graphs were used to construct error-correcting codes in [Tan81, SS96, Spi96]), we only recently discovered that these objects are almost *the same* when interpreted appropriately. We feel that the new perspective gained through this unification makes the time ripe for substantial advances in the area. Moreover, these connections mean that progress on even one aspect of pseudorandomness can propagate and imply improvements throughout the field. With this in mind, the broad goals of this career development plan are the following:

- Understand more fully and strengthen the connections between these various pseudorandom objects.
- Exploit these connections to attack some of the important open problems in the area.
- Make use of the improved understanding of pseudorandomness to improve and extend its applications in other areas, such as cryptography, complexity theory, and data structures.

- Involve students at all levels of the research, from testing potential constructions through computer experiments, as might be suitable for undergraduates, to the mathematical research suitable for Ph.D. students.

- Convey the unified theory of pseudorandomness that we are developing to a larger community. This will be done by developing a new graduate course on the pseudorandomness (with publicly available lecture notes) and filtering aspects of the theory into undergraduate and graduate courses on related topics.

Institutional Context. Harvard provides an ideal setting to carry out the above career development setting. First, the strong core Theory of Computation group will provide a source of stimulation as I explore the connections of pseudorandomness to other topics. Specifically, I expect to have such interactions with Leslie Valiant (on circuit lower bounds), Michael Rabin (on cryptography), and Michael Mitzenmacher (on data structures). I am also likely to interact with members of Harvard's renowned mathematics department, as the evidence increases that my work on expander graphs [RVW00] may have significance for deep questions in algebra (cf., [ALW01, Gro00] and Section 3.1). I may also benefit from the presence of several experts on error-correcting codes at Harvard, including Michael Mitzenmacher (computer science), Noam Elkies (mathematics), and Alek Kavcic (electrical engineering). Such connections will be facilitated by the fact that Computer Science at Harvard is not separated from the rest of the university by the administrative walls of a department, but is rather part of a larger interdisciplinary structure known as the Division of Engineering & Applied Sciences.

In addition, the university is very supportive of my career development plans. They provided me with a generous start-up package that has enabled me to hire an outstanding postdoc, Eli Ben-Sasson, for the 2001–02 academic year. I hope that the funds resulting from this proposal will help me keep Eli at Harvard for one more year, so we can continue to collaborate on this topic. Harvard has also given me great freedom in terms of teaching, which I will use to design new graduate and undergraduate courses related to this proposal (as described in Section 4).

2 Previous Work — The Connections

2.1 Extractors and Pseudorandom Generators

For nearly a decade, work on pseudorandom generators and work on extractors were viewed as orthogonal directions in the theory of pseudorandomness. This is because, in contrast to extractors, pseudorandom generators are complexity-theoretic objects. In order to construct them, we make some assumption, like we have a function f which is hard to compute, and then we construct a pseudorandom generator from f. Like most work in complexity theory, we only know how to prove the correctness of such a construction via a *reduction*. Namely, we show that if we had an efficient algorithm which could distinguish the output of the pseudorandom generator from truly random, then we could transform it into an efficient algorithm to compute f, contradicting our assumption. In contrast, extractors are information-theoretic objects. Constructions of extractors make no complexity assumptions, and typically their correctness is proved via probabilistic arguments rather than reductions.

Despite these apparent differences, Trevisan [Tre99] established a surprising connection between extractors and pseudorandom generators. He showed that any construction of pseudorandom generators of the above form (which works using "any" hard function f, and whose correctness is proved by a "black-box" reduction) *is also* a construction of extractors, when interpreted appropriately. In a sense, he showed that extractors are the "information-theoretic analogue" of pseudorandom generators. Using this connection, he gave a strikingly simple extractor construction based on the pseudorandom generator construction of Nisan and Wigderson [NW94], and the parameters of his extractor improved over many previous constructions.

This connection strongly suggests that there should be a *single* construction which yields both optimal extractors and optimal pseudorandom generators for most ranges of parameters. However, despite significant improvements to Trevisan's construction by myself and others [RRV99b, ISW00, TUZ01], an optimal construction has remained elusive.

2.2 Extractors and Expander Graphs

It has been known since the original paper of Nisan and Zuckerman [NZ96] which defined extractors that an extractor could be viewed as a certain kind of unbalanced, bipartite expander graph, and indeed Wigderson and Zuckerman [WZ99] used known constructions of extractors to give constructions of undirected expander graphs that "beat the eigenvalue bound." However, in these constructions, the degree of the graph grows as a function of the number of vertices. Expander graphs of *constant degree* are of particular importance, but these were viewed as different and more difficult than extractors. Indeed, the only known explicit constructions of constant-degree expander graphs relied on sophisticated algebraic techniques [Mar73, GG81, AM85, AGM87, JM87, LPS88, Mar88, Mor94], whereas works on extractors emphasized that in contrast they only used "elementary techniques."

Recently, Omer Reingold, Avi Wigderson, and I [RVW00] pointed out that the connection between extractors and expanders does indeed extend to the *constant-degree case*. Specifically, the case of constant-degree expanders corresponds to case of "high min-entropy" extractors, which is typically viewed as the *easiest* range of parameters. We reconciled these contradictory viewpoints (namely, the "difficulty" of constant-degree expanders and the "easiness" of high min-entropy expanders). Specifically, we gave a simple construction of nearly optimal high min-entropy extractors, and, by translating the construction and proof to the setting of expanders, obtained a new construction of constant-degree expanders. These expanders are obtained by starting with a simple "constant size" base expander and repeatedly applying our new *zig-zag graph product* to it. The resulting construction is combinatorial (as opposed to algebraic) and follows a clear intuition; having such a construction of expanders was a long-standing open problem. More importantly, it gives us a new, powerful way of reasoning about expander graphs using the probabilistic language of extractors, and makes several of the open problems about constant-degree expander graphs seem less daunting.[1]

2.3 The Use of Error-Correcting Codes

The pseudorandom generator construction of Nisan and Wigderson [NW94] mentioned earlier requires starting with a Boolean function f that is very *hard on average*. That is, for a random input x, no efficient algorithm can compute $f(x)$ much better than random guessing (i.e., the success probability is at most $1/2+\varepsilon$, for small ε). In later works, error-correcting codes were used to weaken this assumption to one which just requires that f is hard in the *worst case* (i.e., no efficient algorithm can can compute $f(x)$ correctly for all x).

The original idea dates to the work of Babai, Fortnow, Nisan, and Wigderson [BFNW93], who showed that if we encode a worst-case hard Boolean function f as a low-degree multivariate polynomial $\hat{f}$, then $\hat{f}$ is somewhat hard on average. The reason for this is as follows: if there were an efficient algorithm A that could correctly compute $\hat{f}$ on most inputs, then an efficient decoding algorithm for multivariate

[1] For example, already in our paper we give a nearly optimal construction of graphs in which every two subsets of density $1/A$ are connected (for constant or slowly growing A); our graphs have degree $O(A \cdot \mathrm{polylog}\, A)$ which is substantially closer to the slower bound of $O(A \cdot \log A)$ than previous constructions (which had degree either growing with the number of vertices [WZ99] or degree $\Omega(A^2)$ achieved by Ramanujan graphs [LPS88, Mar88]).

polynomials [Lip89, BF90] could be used to convert A into an efficient algorithm A' which computes f correctly everywhere. Unfortunately, using the classical notion of decoding, this does not prove that $\hat{f}$ is sufficiently hard-on-average for the Nisan–Wigderson pseudorandom generator construction. (This argument only proves that no algorithm can correctly compute $\hat{f}$ on *most inputs*, rather than just slightly better than random guessing.) Instead, Yao's XOR Lemma [Yao82] was used to "amplify" the hardness of $\hat{f}$ to the required level, but this introduced inefficiency into the construction, and substantial work was done to "derandomize" the XOR Lemma and remove these inefficiencies [Imp95, IW97]. In [STV01], Madhu Sudan, Luca Trevisan, and I proved that actually the polynomial encoding $\hat{f}$ is already as hard-on-average as needed, by using a *list*-decoding algorithm for low-degree multivariate polynomials (which was first given in [AS97], but we also provided a simpler and quantitatively better algorithm). Indeed, this gave an optimal conversion from worst-case hardness to average-case hardness, and showed that a pseudorandom generator could be built by just combining an error-correcting code with the Nisan–Wigderson construction.

When Trevisan established his connection between pseudorandom generators and extractors [Tre99], his extractor construction inherited the same structure — the use of an error-correcting code followed by the Nisan–Wigderson construction (which actually just projects symbols from cleverly selected positions in the codeword). Recently, Ta-Shma and Zuckerman [TZ01] showed that this use of error-correcting codes is *not* a coincidence — when viewed appropriately, extractors are a generalization of list-decodable error-correcting codes. This suggests that error-correcting codes should play a more central role in pseudorandom generator and extractor constructions, and that better constructions may be possible if we exploit the structure of specific error-correcting codes (such as multivariate polynomials) rather than use the Nisan–Wigderson construction. Remarkably, Ta-Shma, Zuckerman, and Safra [TZS01] recently managed to succeed in building extractors directly from multivariate polynomial codes, and their proof of correctness proceeds via a (very nontrivial) generalization of our list-decoding algorithm from [STV01]. Shaltiel and Umans [SU01] have substantially improved the construction and shown how to also obtain a pseudorandom generator from it. While these constructions do not improve upon all previous extractor constructions in terms of parameters, their coding-theoretic approach seems the most promising for ultimately obtaining optimal constructions of both extractors and pseudorandom generators.

3 Proposed Research

This proposal is about the exciting possibilities raised by the connections described in the previous section. They signify a deeper understanding of pseudorandomness, that we plan to cultivate over the course of this research. We believe that, over the next five years, we can use these connections to make significant progress toward resolving some of important open problems in the area. We will also extend our investigations to some of the applications of pseudorandomness to other areas such as cryptography and data structures.

Below we give a few examples of specific questions on which we hope to make progress, and sketch our new approaches for attacking them. We stress that this is not intended to be a comprehensive list of everything we will do in the course of this work. Rather, this list is intended to give a flavor of the type of progress we believe we can make with the newly discovered connections in pseudorandomness. As with any good research, we hope that exciting discoveries will take us in unexpected directions.

3.1 Expander Graphs

We begin by describing research problems about expander graphs which we will address using the connection between extractors and expander graphs, and in particular, our new zig-zag graph product [RVW00]. Much of the work described below will be done in collaboration with **Omer Reingold** and **Avi Wigderson**.

Constant-Degree Expanders with Near-Optimal Expansion. Earlier, we informally described expander graphs as *sparse* graphs that are nevertheless *highly connected*. Sparsity is typically measured by the degree of the graph, but there are several different measures of *expansion*, i.e. how "highly connected" a graph is. The classical measure, called *vertex expansion*, requires that there are constants $c > 1$ and $\alpha < 1$ (independent of the number of vertices) such that for every subset S of vertices of density at most α, the neighborhood of S is of size at least $c \cdot |S|$. c is called the *expansion factor* of the graph, and the relationship between c and the degree d of the graph is of fundamental importance. In a random graph, it is known that $c = d - 1$ can be achieved (for some constant α depending on d). However, explicit constructions have failed to match this bound. Indeed, it was considered a breakthrough to even have c and d be constants independent of the number of vertices [Mar73, GG81]. One reason that achieving near-optimal expansion was elusive is that in almost all of the explicit constructions, the expansion property was proved by bounding the second largest eigenvalue λ of the graph's adjacency matrix, which was known to imply a lower bound on the expansion constant c [Tan84]. However, it was shown by Kahale [Kah95] that a bound on λ alone is *insufficient* to prove that $c > d/2$. This is unfortunate, because many of the computer science applications of expander graphs require expansion greater than $d/2$ [Spi96, ALM96, BMRS00] (and thus, to date, these applications have had to remain nonconstructive.)

We believe that, using our new zig-zag graph product, it should be possible to achieve expansion $(1-\varepsilon)\cdot d$ for an arbitrarily small constant ε. The reason is that our zig-zag graph product can be analyzed using several different measures of expansion, some of which are not subject to the $c \leq d/2$ limitation of the eigenvalue measure. Specifically, we propose the notion of a *min-entropy expander* which we proceed to explain in more detail. The *min-entropy* of a discrete probability distribution X is defined to be $\mathrm{H}_\infty(X) \overset{\mathrm{def}}{=} \log_2 (1/\max_x \Pr[X = x])$. Thus, if X is supported on a set S, $\mathrm{H}_\infty(X) \leq \log_2 |S|$, with equality iff X is uniform on S. Now, for a graph G and a probability distribution π on the vertices of G, let $G\pi$ denote the probability distribution induced by selecting a vertex of G according to π and then moving to a random neighbor of that vertex. We call G a (k, a, ε) *min-entropy expander* if for every distribution π of min-entropy at most k, the distribution $G\pi$ is ε-close (in statistical difference) to some distribution with min-entropy $\mathrm{H}_\infty(X) + a$. In particular, all sets of size at most 2^k expand by a factor at least $(1 - \varepsilon) \cdot 2^a$. So, if we construct min-entropy expanders of constant degree d on a growing number n of vertices with $k = \log_2 n - O(1)$ and $a = \log_2 d$, then we will have constant-degree expanders with expansion $(1 - \varepsilon) \cdot d$ for all sets of up to constant density.

This notion of min-entropy expander is a slight generalization of the various notions of "condensers" recently considered in the extractor literature [RR99, RSW00, TUZ01]. In particular, the observation that "loss-less" condensers (i.e., min-entropy expanders with $a = \log_2 d$) are in fact graphs with expansion $(1 - \varepsilon)d$ is due to [TUZ01]. They actually constructed such graphs for nonconstant degree $d = \mathrm{polylog}\, n$. We hope to obtain a similar result for *constant* degree d via two steps. First, we need to analyze our zig-zag-based expander construction in terms of min-entropy expansion. In [RVW00], we analyze the zig-zag product in two ways: we analyze its effect on the second largest eigenvalue, which essentially measures how a random walk increases Renyi's 2-entropy (as opposed to min-entropy); and we analyze its effect on extraction, which is similar to min-entropy expansion except that distributions of min-entropy *at least* k are considered and we ask that $G\pi$ is ε-close to *uniform*. Since min-entropy expansion is a blend of these two measures, we are optimistic that the analysis can be translated. Second, in order to actually get expansion $(1 - \varepsilon)d$ (rather than just $\Omega(d)$), we hope to use an idea of Raz and Reingold [RR99], which shows how hash functions with low collision probability can be used to transform extractors into loss-less condensers. A direct application of their idea increases the degree of the graph by a $\mathrm{polylog}\, n$ factor, but we hope that applying their idea *within the zig-zag product*, we may be able to only incur a constant factor penalty.

Further Properties of the Zig-Zag Product. Our zig-zag graph product is novel among graph products in that it preserves both sparsity and expansion while increasing size. Though our analysis of it in [RVW00] was already sufficient to give a new construction of constant-degree expanders, there is still much that we do not understand about it. Below I enumerate just a few of the natural questions that arise:

1. The way we construct an infinite family of constant-degree expanders is to start with a constant-size base graph H, and then repeatedly apply the zig-zag product, alternating with standard graph operations such as squaring and tensoring the adjacency matrix. We do not know whether applying the other graph operations (squaring and tensoring) are really necessary, or just a deficiency in our analysis. Can we get an infinite family of expanders by repeatedly applying *just* the zig-zag product to some (or any) base graph H? A positive answer to this question would greatly improve the computational efficiency of the resulting expanders (neighborhoods can be computed in constant space and linear time, rather than just polynomial time), which is essential for some applications (e.g., [BGW99]).

2. It is known that the optimal bound on the second largest eigenvalue of an infinite family of degree d graphs is $\lambda = 2\sqrt{d-1}$ [LPS88, Mar88, Nil91], and graphs meeting this bound are referred to as *Ramanujan*. Using variants of the zig-zag graph product, the best we can currently achieve is $\lambda = O(d^{2/3})$. Can we construct Ramanujan graphs using some variant of the zig-zag product?[2]

3. Can we understand the eigenvectors of the zig-zag product in terms of the eigenvectors of the original graphs, and interpret them combinatorially? Our analysis bounds the second largest eigenvalue. The eigenvectors themselves can give deeper insight into the effect of the product on expansion.

4. How does the zig-zag graph product (and variants) affect other graph properties such as girth, diameter, chromatic number?

The above questions are particularly appealing from an educational perspective, because computer experimentation can play a major role in the research. Specifically, undergraduates can write programs to perform the zig-zag graph product and calculate properties of the resulting graphs (such as the second largest eigenvalue). Such experiments can help formulate conjectures, which we can subsequently try to prove (possibly using additional insight gained from the experiment). I believe this is an ideal way to give undergraduates a gentle introduction to theoretical research. It gives them exposure to difficult theoretical research questions, while still enabling them to obtain some tangible results (instead of facing the usual frustration of not knowing where to begin). I intend to actively involve undergraduates in this research, and in any other aspects of this project where similar opportunities arise.

A Connection to Algebra. As mentioned earlier, one of the original motivations of our work [RVW00] on the zig-zag graph product was to have a non-algebraic construction of expander graphs. Nevertheless, subsequent to our work, Alon, Lubotzky, and Wigderson [ALW01] have shown a surprising connection between the zig-zag graph product and algebra. Specifically, they showed that, under certain algebraic conditions, taking the zig-zag product of the Cayley graphs of two finite groups G_1 and G_2 yields a Cayley graph for the *semidirect product* of the groups G_1 and G_2. Using this connection with our analysis of the zig-zag product, they disproved an earlier conjecture of Lubotzky and Weiss [LW93] which asserted that the property of a (constant-degree) Cayley graph being an expander depends only on the group (and not on the choice of generators).

[2] In the previous section, we argued that the eigenvalue is *not* an ideal measure of expansion if one is ultimately interested in vertex expansion, as is often the case. However, bounds on the eigenvalue are in some ways stronger than vertex expansion, and a significant number of applications of expanders directly make use of the eigenvalue measure.

Given the importance of expanding Cayley graphs in mathematics (cf., [Gro00, LP01]), this connection merits a deeper investigation. One hope is that, using with the zig-zag product (or variants), we can construct infinite families of expanding Cayley graphs with more control over their algebraic structure than permitted by previous constructions. Perhaps this can even lead to constructions of Cayley graphs of the kind needed for Gromov's work on the Baum-Connes conjecture [Gro00].

3.2 Derandomization

One of the primary applications of pseudorandomness is to address the question: *Are randomized algorithms more powerful than deterministic ones?* That is, how does randomization trade off with other computational resources? Can every randomized algorithm be converted into a deterministic one with only a polynomial slowdown (*i.e.,* does $\mathbf{P} = \mathbf{BPP}$)? or with only a constant-factor increase in space (*i.e.,* does $\mathbf{L} = \mathbf{RL}$)? Both of these questions have been the subject of a large and exciting body of research, and completely resolving them seems quite difficult. Still, we believe that we have interesting new approaches that have a good chance of shedding new light on these important problems.

Derandomization in the Uniform Setting. The only known approach for efficiently converting arbitrary randomized algorithms to deterministic ones is to construct pseudorandom generators. For time-bounded algorithms (*e.g.,* $\mathbf{BPP}$), we only know how to construct pseudorandom generators based on complexity assumptions. As described in Section 2, pseudorandom generators can be constructed from "hard" functions, and in this area, "hard" typically means having high *circuit complexity*. To make the pseudorandom generator itself computationally efficient, we typically also need to assume that the hard function lies in $\mathbf{EXP}$ (i.e., is computable in exponential time). Since proving unconditional circuit lower bounds for $\mathbf{EXP}$ seems well beyond current techniques, it is natural to ask: *Do we really need circuit lower bounds for derandomization?*

Impagliazzo and Wigderson [IW98] have given some indication that the answer to this question is NO. Specifically, they showed that some nontrivial derandomization of $\mathbf{BPP}$ is possible merely under the "uniform" assumption that $\mathbf{EXP} \neq \mathbf{BPP}$. The heart of their work is to show that if the hard function used has some additional nice properties — namely "downward self-reducibility" and "random self-reducibility" — then only uniform hardness is needed. To obtain these properties, they argue that they can assume that their hard problem is the PERMANENT using lots of machinery (a version of the Karp-Lipton Theorem [KL82], Valiant's Theorem [Val79], Toda's Theorem [Tod91], the $\mathbf{MIP}$ characterization of $\mathbf{EXP}$ [BFL91]). While the final theorem is a great result, this two-step proof is dissatisfactory in a couple of ways. The use of all the machinery makes it less clear why the final result is true. It also causes quantitative inefficiency which prevents the establishment of a true randomness-time trade-off. For example, their technique does not give a *polynomial-time* derandomization of $\mathbf{BPP}$ under strong enough uniform assumptions (while it is possible under a strong enough *circuit complexity* assumption [IW97]). If the quantitative inefficiencies could be removed, it would also be possible to unconditionally prove something in the spirit of "For every nice function t, either $\mathbf{EXP} \subseteq \mathbf{BPTIME}(t(n))$ or $\mathbf{BPTIME}(t(n)) \subseteq \mathbf{EXP}$,"[3] i.e. all probabilistic time classes are *comparable* with $\mathbf{EXP}$. Instead, their techniques only imply something like "For every nice function t, either $\mathbf{EXP} \subseteq \mathbf{BPTIME}(t(t(n)))$ or $\mathbf{BPTIME}(t(n)) \subseteq \mathbf{EXP}$," which becomes vacuous for many interesting values of t (e.g. $t(n) = 2^{n^\varepsilon}$).

In ongoing joint work with Luca Trevisan, we have made progress on resolving these deficiencies. First, using ideas from the proof of $\mathbf{IP} = \mathbf{PSPACE}$ [LFKN92, Sha92], we can directly exhibit a random self-reducible and downward self-reducible complete problem for $\mathbf{PSPACE}$, which eliminates the use Valiant's Theorem and Toda's Theorem from the proof. Second, we have completely solved the quantitative

[3] This is an oversimplification. For readability, many technical qualifications (e.g. polynomial slackness factors, and inclusions being only "infinitely-often average-case simulations") are omitted.

inefficiencies for a significant part of the pseudorandom generator construction, namely the conversion from worst-case hardness to average-case hardness (as described in Section 2.3). In order to explain the latter in more detail, let us recall the idea behind worst-case to average-case conversion based on error correcting codes. We start out with a worst-case hard function f, and consider its encoding $\hat{f}$ in an error-correcting code. Then we argue that if there were an efficient algorithm which computes $\hat{f}$ on average, we could use the decoding algorithm to obtain an efficient algorithm computing f everywhere. The problem is that in order to obtain sufficiently hard-on-average problems, we must work with parameters where unique decoding is no longer possible, and only *list decoding* is possible. But how do we know which candidate in the "list" is the correct decoding? In the circuit complexity setting, this is solved by "hard-wiring" that information into the circuit. To solve this problem in the uniform setting, we first observe that our work in [STV01] actually gives a *uniform* algorithm for generating the list of decodings. Next, we observe that the $\mathbf{MIP}$ characterization of $\mathbf{EXP}$ implies that we can assume that the hard function f has a "self-testability" property (cf., [BFL91]), which we can use to "prune" the list of possible decodings to a single correct candidate which is correct almost everywhere (and then we can "self-correct" it to make it correct everywhere).

It seems significantly more challenging to extend these ideas to an entire pseudorandom generator construction in the uniform setting, but the progress we have already obtained gives us optimism about our coding-theoretic viewpoint. Perhaps ideas from the coding-theoretic extractor and pseudorandom generator constructions of [TZS01, SU01] will help here.

While above we discuss approaches to derandomization without circuit complexity assumptions, an opposite research direction is to show that circuit complexity assumptions are *necessary* for some derandomization results. The first such theorem was recently obtained by Impagliazzo, Kabanets, and Wigderson [IKW01] in the "nondeterministic" setting, who showed that a nontrivial derandomization of $\mathbf{MA}$ (a probabilistic version of $\mathbf{NP}$) is possible if and only if $\mathbf{NEXP}$ (nondeterministic exponential time) has problems of high circuit complexity. What about $\mathbf{BPP}$? It is well-known that if $\mathbf{BPP} = \mathbf{P}$ for "promise problems" then $\mathbf{MA}$ can be fully derandomized (cf., [GZ97]), which implies circuit lower bounds for $\mathbf{NEXP}$ by the aforementioned result. But does it imply circuit lower bounds for $\mathbf{EXP}$? A positive answer would show that derandomizing $\mathbf{BPP}$ is *equivalent* to proving circuit lower bounds for $\mathbf{EXP}$ (since circuit lower bounds for $\mathbf{EXP}$ suffice to derandomize $\mathbf{BPP}$ [BFNW93]).

It is interesting that these recent developments on derandomization make essential use of results on *interactive proof systems* (such as the $\mathbf{MIP}$ characterization of $\mathbf{EXP}$). Since I have done a great deal of research on interactive proofs [SV97, SV99, GSV98, GV99, GSV99, Vad00, Vad00, GVW01], including my Ph.D. thesis [Vad99], I am in a unique position to work on the interplay between these topics.

Derandomizing Space-Bounded Computation. (This work will be done in collaboration with **Omer Reingold** and **Ronen Shaltiel**.) One of the most basic algorithmic questions is whether connectivity in an undirected graph can be decided in space $O(\log n)$. If we allow randomization, the answer is YES [AKL$^+$79], but for deterministic algorithms it is a long-standing open problem. The main approach to this problem is through pseudorandom generators against space-bounded algorithms, as initiated in [AKS87, BNS89, Nis92]. In contrast to the time-bounded case, pseudorandom generators against space-bounded algorithms can be constructed without making any assumptions. But despite a large body of further work [NZ96, INW94, SZ99, Arm98, ATWZ00, RR99], the known pseudorandom generators appear insufficient to obtain a deterministic log-space algorithm for undirected connectivity. The best deterministic space bound is $O(\log^{4/3} n)$ for undirected connectivity, due to Armoni, Ta-Shma, Wigderson, and Zhou [ATWZ00], and $O(\log^{3/2} n)$ for general randomized log-space algorithms, due to Saks and Zhou [SZ99].

The starting point for our research is a pseudorandom generator construction of Impagliazzo, Nisan, and Wigderson [INW94], as modified by Raz and Reingold [RR99]. Suppose we want a pseudorandom generator which, from a short seed, generates R bits that look random to any algorithm which runs in space

S. The INW pseudorandom generator is a recursive construction which, at each level, roughly doubles its output length. Thus, the depth of the recursion is $\Theta(\log R)$. Each level of recursion increases the seed length by $\Theta(S + \log(1/\varepsilon))$ bits, where ε is an error parameter, for a total seed length of $\Theta(\log R \cdot (S + \log(1/\varepsilon))$. In the analysis, the errors accumulate linearly with the number of random bits generated, so we must take $\varepsilon < 1/R$. This gives a total seed length of $\Theta((S + \log R) \cdot \log R)$, which is the same as Nisan's original generator.

To do better, argued Raz and Reingold [RR99], we should try to reduce the cost of $\Theta(S + \log(1/\varepsilon))$ incurred at each level of recursion. They focused on eliminating the increase due to the space S, and succeeded in doing so under the additional assumption that weak estimations to the state probabilities of the space-bounded algorithm can be computed. (This assumption is more or less satisfied by the randomized log-space algorithm for undirected connectivity.) This yields a pseudorandom generator with seed length $\tilde{O}(\log^2 R)$, which is a substantial improvement when R is significantly smaller than 2^S.

Our proposal is to focus on the other cost at each level of recursion, namely the $\Theta(\log 1/\varepsilon)$ bits due to the error parameter ε. Before describing our approach, let us see the potential impact. If that cost could be completely eliminated, the total seed length would be $O(S \log R)$. At first, this seems to be no better than $O((S + \log R) \cdot \log R)$, because one typically assumes that a space S algorithm cannot use more than 2^S random bits, for otherwise there are infinite loops. But this is not quite true, since the space bound S only refers to the number of bits stored that depend on the random bits, and does not exclude, for example, a time counter which prevents infinite loops. In particular, when S is constant, a pseudorandom generator with seed length $O(S \log R) = O(\log R)$ would imply a full derandomization of read-once constant-width branching programs, which includes the well-studied special case of constructing discrepancy sets for combinatorial rectangles [LLSZ97, ASWZ96]. Even more optimistically, if our technique succeeds and could be combined with the one of Raz and Reingold [RR99], the result could conceivably be an$\tilde{O}(\log n)$-space algorithm for undirected connectivity.

Now we describe our approach to dealing with the $\Theta(\log(1/\varepsilon))$ cost. The new observation is that the pseudorandom generator still produces something nontrivial even when ε is taken to be, say, $\Theta(1/\log^2 R)$, rather than $\Theta(1/R)$. Although we cannot prove the output to be pseudorandom, we can prove that each bit of the output cannot be predicted from the previous bits with probability more than $1/2 + 1/\log R$ by any space S algorithm. Thus we "only" need to convert this mild unpredictability into stronger unpredictability. A natural approach to doing so is to use Yao's XOR Lemma [Yao82]: run the generator on k independent seeds and take the exclusive-OR of the outputs. Intuitively, this should reduce the predictability of each bit to roughly $1/2+1/\log^k R$. There are two problems with this approach. The first is that existing proofs of Yao's XOR Lemma typically pay a substantial price in the efficiency of the predictor, whereas we cannot afford more than a constant-factor loss in the space. To resolve this problem, we hope to exploit a recent work of Shaltiel [Sha01], in which he proves an XOR Lemma for certain communication complexity problems where the communication bound increases rather than decreases. Since randomized space-bounded computation is closely related to communication complexity, there is some hope that the ideas might translate. A second problem is that, even if the XOR Lemma works out perfectly, the seed length increases by a factor of k, and we would end up with nothing better than before. To solve this, we would need to derandomize our XOR Lemma, as has previously been done in the time-bounded case in [Imp95, IW97]. Needless to say, there are many places this approach can go wrong. Still we feel that it is an interesting one, and there is some chance that it will yield progress on this important problem.

3.3 Other Connections

As mentioned earlier, pseudorandomness has applications to other areas of computer science, and we expect our research to naturally branch out into some of these directions. Below we list a few specific possibilities.

Pseudorandomness in Cryptography. The modern theory of pseudorandomness was actually initiated by Blum, Micali, and Yao (BMY) [BM84, Yao82] with cryptography in mind. Indeed, the notion of pseudorandom generator they proposed and its generalization to *pseudorandom functions* [GGM86] immediately solves most of the problems of private-key cryptography. The BMY notion of pseudorandom generator has one important difference from the one we have been discussing for the bulk of this proposal, which was proposed by Nisan and Wigderson (NW) [NW94]. Namely, the NW definition allows that the pseudorandom generator has a (slightly) greater running time than the algorithms it must fool (which is sufficient for the application to derandomization), whereas the BMY definition requires that the pseudorandom generator even fool algorithms that have more running time than the generator (which is essential in cryptographic applications, and also other applications such as to learning theory [Val84] and limitations on circuit lower bounds [RR97]).

While the proposed research primarily involves the NW notion of pseudorandom generators and its connections to other pseudorandom objects, we have hope that the research can be linked back to BMY notion and cryptography. Indeed, I have done a significant amount of research on cryptography (e.g., [GSV98, BHSV98, BGI$^+$01]) and even specifically on pseudorandomness in cryptography [MRV99]. If a link to the BMY notion can be established, the benefits could go in both directions. One hope is that some of the ideas that have been developed in this unified theory of pseudorandomness can be used to give more efficient constructions of pseudorandom generators in the BMY sense. In particular, a long-standing open problem is to give a more efficient construction of BMY-type pseudorandom generators from any one-way function; the only known construction is in the celebrated work of Håstad, Impagaliazzo, Levin, and Luby [HILL99], which is complicated, inefficient, and impractical. In the other direction, a hope is that Trevisan's connection between pseudorandom generators and extractors can be extended to use the BMY construction of pseudorandom generators from one-way permutations (rather than the NW-style constructions of pseudorandom generators from hard Boolean functions). To carry this out, we would need a worst-case to average-case connection for permutations. Specifically, from any permutation π which is hard to invert in the worst case, can we construct a permutation $\hat{\pi}$ such that $\hat{\pi}$ is hard to invert *on average* even for algorithms that have oracle access to $\hat{\pi}$ in the forward direction? Perhaps some generalization of our coding-theoretic approach to worst-case/average-case conversion for Boolean functions [STV01] can be made to work.

Randomized Data Structures. Pseudorandomness also turns out to be quite useful in various data structure problems. For example, tools from pseudorandomness, such as small families of hash functions and expander graphs, have proven to be useful in constructing data structures for efficiently storing a small set S of elements from a large universe $[N]$ (cf., [FKS84, BMRS00]). One variant of this problem is the *approximate set storage problem*, where the data structure is produced from S in a probabilistic manner which guarantees that for every element $x \in [N]$ of the universe, the data structure will allow one to determine whether $x \in S$ with some small, controllable error probability (taken over the probabilistic construction of the data structure). A *Bloom Filter* [Blo70] is a construction which allows one to save substantially in the size of the data structure as compared to storing the set exactly (specifically, using $O(|S|)$ bits rather than $O(|S| \log N)$). Unfortunately, Bloom Filters, as presented in the literature, are nonconstructive because they assume access to a completely random hash function, which cannot even be described without exponential storage. Instead, one should use a small, explicit family of hash functions which can be generated (and hence stored) using just a few random bits, while still permitting the analysis of Bloom Filters to hold. Doing this is essentially a derandomization problem, and thus our expertise in pseudorandomness can be very helpful in finding an optimal solution. We plan to work on this and other problems about randomized data structures with **Michael Mitzenmacher**, who has done work on Bloom Filters in the past [Mit01].

Pseudorandomness and Circuit Lower Bounds. Proving strong circuit lower bounds for natural functions is perhaps the most difficult open problem in computational complexity. Given that proving even superlinear size lower bounds for general circuits seems well out of reach, researchers have turned to proving lower bounds on other "easier" resources, such as space (branching program size) and parallel time (circuit depth). For both of these problems, extractors seem like they could be useful. Consider a random input to a small-space algorithm. At "most" states of the algorithm, the algorithm "remembers" very little information about the input and thus the input still has a lot of "randomness" left. Hence, by the definition of an extractor, the algorithm cannot distinguish the output of an extractor applied to this input from truly random. This suggests that it may be possible to prove a branching program lower bound for some function related to extractors. A similar approach also seems compelling for finding an explicit function which cannot be computed by linear-size log-depth circuits, via Valiant's work [Val77]. Of course, the above intuition does not directly work: extractors make use of a short random seed, and the above reasoning does not apply once the small-space algorithm reads the seed. But still, given our rapidly improving understanding of extractors, perhaps something along these lines can be made to work.

4 The Educational Component

In addition to helping advance our understanding of computation through research, my career goals also include sharing exciting ideas in computer science with people outside or just entering the research community. For this reason, I intend to make teaching and advising a top priority throughout my career.

A Graduate Course on Pseudorandomness. During the Spring 2002 term, I will teach a new graduate course at Harvard entitled *Pseudorandomness*. The course will examine many of the pseudorandom objects discussed in this proposal — pseudorandom generators, expander graphs, extractors, error-correcting codes — and their applications in areas such as cryptography, complexity theory, combinatorics, and data structures. The theory of pseudorandomness has grown into one of the most active research areas in theoretical computer science with connections to a wide variety of areas, so it is important to make this material accessible to graduate students, and not just those who wish to do research on the subject. However, most of the relevant material has not made its way into textbooks except in scattered bits. (An exception is the recent monograph of Goldreich [Gol99], but that is written more survey-style than textbook-style with most details omitted.) In the past few years, several people at other universities have developed graduate courses on this topic, but what will make my course unique is the strong emphasis on the *unified theory* that is developing via the connections described in Section 2. During the course, we (the students and myself) will produce a set of lecture notes which I will make publicly available. I expect to teach this course every second or third year at Harvard, and each time refining the lecture notes and the course itself.

An Undergraduate Course on Cryptography. In the Fall 2001 term, I will introduce the first undergraduate course on cryptography at Harvard, entitled *Introduction to Cryptography*. Despite the great importance of cryptography in today's electronic economy, there are few true cryptography courses for undergraduates in existence. Most attempts are instead "network security" courses which abandon the modern, rigorous approach to cryptography that has developed over the past two decades, probably under the assumption that it is "too difficult" for undergraduates. I disagree with this assumption; I believe that the basic principles guiding the modern approach (the careful approach to definitions, the meaning of "provable security") *can* be conveyed to undergraduates. Moreover, I believe this is important also for practical reasons. Undergraduate students are ultimately the people who will go into industry and implement cryptographic systems. If all the dangers of a non-rigorous approach to cryptography remain confined to graduate courses, they will never work their way into practice. This is not to say that my course will be an entirely theoretical course. It will also contain material on how cryptography is actually used (and misused) in practice, and also how cryptography fits into the larger contexts of network and systems security.

What does this have to do with pseudorandomness? I believe that one way to convey the foundations of cryptography without a dizzying array of definitions and constructions is to use pseudorandomness as a common theme tying the different notions together. Specifically, one can begin with private-key encryption, and introduce pseudorandom generators as a way to generate a "computational" one-time pad from a short, shared random key (thereby circumventing Shannon's limitations on information-theoretic security [Sha49]). More efficient constructions, based on pseudorandom functions and block ciphers, can be presented as generalizations of this same principle. Similarly, the Blum-Goldwasser [BG84] construction of *public-key* encryption schemes is also based on the same idea. Message authentication can be explained in terms of *unpredictability*, which in turn can be presented as a natural weakening of pseudorandomness.

I expect to teach this course for at least several years, until the curriculum has become sufficiently polished that someone else can easily take over.

Other Courses. When I am not teaching the above courses, I will have opportunities to teach courses on other areas, e.g. undergraduate courses such as *Introduction to the Theory of Computation*, *Graph Theory and Combinatorics*, or *Data Structures and Algorithms*, and graduate courses such as *Computational Complexity* and *Cryptography*. Some aspect of each of these courses relates to the theory of pseudorandomness, and I hope to introduce some ideas from pseudorandomness into these courses when I teach them.

Student Involvement in Research. Students will play a major part in the research of this proposal. I plan to build a group of three or four Ph.D. students, of which I expect to have two working on the research in this proposal. One student, named Minh Nguyen, has already begun working with me. I also plan to involve undergraduates in the research, by finding accessible projects for them along the lines of the computer experiments described in Section 3.1. In September 2001, an outstanding postdoc, named Eli Ben-Sasson, is coming to Harvard to work with me on pseudorandomness and other topics. During the 2001–02 academic year, I will support him using the generous start-up package provided to me by Harvard. For the following year (i.e., the first year of this proposal), I budgeted 50% support for him so that he can stay and continue collaborating with me on this topic. Finally, now that the Theory of Computation group at Harvard has sufficient size (4-5 active faculty members, 2 postdocs, 4 students in 2001–02), I plan to organize a weekly Theory of Computation seminar. This will be a forum for me and my students to convey our work on pseudorandomness to other computer scientists and mathematicians in the Boston area. I will make an extra effort to recruit women and minorities as Ph.D. students; indeed, my first student (Minh Nguyen) is a woman.

5 Past Accomplishments

Due to space constraints, I limit myself to describing results from prior NSF support and prior educational accomplishments, Thus, I do not describe my most of my work on interactive and zero-knowledge proofs [SV97, SV99, GSV98, GV99, GSV99], including my Ph.D. thesis [Vad99] which won the *ACM Doctoral Dissertation Award* and the *MIT EECS Sprowls Award*. I also do not describe some of my other work on pseudorandomness [RRV99b, RRV99a, GVW00], cryptography [BHSV98, MRV99], the complexity of counting [Vad95, Vad97], and other topics [BFR$^+$98].

5.1 Results from Prior NSF Support

From September 1999 to December 2000, I was supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship. That work resulted in the following five publications (all of which were presented at least once at international conferences, plus at other workshops and universities):

[Vad00] On transformations of interactive proofs that preserve the prover's complexity (STOC '00). Goldwasser and Sipser [GS89] proved that every interactive proof system can be transformed into a public-coin one (a.k.a., an Arthur–Merlin game). Their transformation has the drawback that the computational complexity of the prover's strategy is not preserved. We show that this is inherent, by proving that the same must be true of any transformation which only uses the original prover and verifier strategies as "black boxes."

[RVW00] Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors (FOCS '00, with O. Reingold and A. Wigderson). The main contribution of this work is a new type of graph product, which we call the *zig-zag product*. Taking a product of a large graph with a small graph, the resulting graph inherits (roughly) its size from the large one, its degree from the small one, and its expansion properties from both! Iteration yields simple explicit constructions of constant-degree expanders of every size, starting from one constant-size expander.

A variant of this product can be applied to extractors, giving the first explicit extractors whose seed length depends (poly)logarithmically on only the entropy deficiency of the source (rather than its length) and that extract almost all the entropy of high min-entropy sources. These high min-entropy extractors have several interesting applications, including the first constant-degree explicit expanders which beat the "eigenvalue bound."

[TV00] Extracting randomness from samplable distributions (FOCS '00, with L. Trevisan). Recall that an *extractor* is a procedure which extracts almost uniform bits from a weak source of randomness (i.e., a source of biased and correlated bits). The extraction necessarily uses a small number of additional truly random bits, which can be eliminated by complete enumeration in some, but not all, applications.

Here we consider the problem of *deterministic extraction*, i.e. extracting randomness without any extra truly random bits. Previously, deterministic extraction procedures were known only for sources satisfying strong independence requirements. In this paper, we look at sources which are *samplable*, i.e. can be generated by an efficient sampling algorithm. We seek an efficient deterministic procedure that, given a sample from any samplable distribution of sufficiently large min-entropy, gives an almost uniformly distributed output. We explore the conditions under which such *deterministic extractors* exist.

We observe that no deterministic extractor exists if the sampler is allowed to use more computational resources than the extractor. On the other hand, if the extractor is allowed (polynomially) more resources than the sampler, we show that deterministic extraction becomes possible. This is true unconditionally in the nonuniform setting (i.e., when the extractor can be computed by a small circuit), and (necessarily) relies on complexity assumptions in the uniform setting. Our uniform extractors are based on a connection between deterministic extraction from samplable distributions and hardness against nondeterministic circuits, and on the use of nondeterminism to substantially speed up "list decoding" algorithms for error-correcting codes such as multivariate polynomial codes and Hadamard-like codes.

[GVW01] On interactive proofs with a laconic prover (ICALP '01, with O. Goldreich and A. Wigderson). We continue the investigation of interactive proofs with bounded communication, as initiated by Goldreich and Håstad [GH98]. Let L be a language that has an interactive proof in which the prover sends few (say b) bits to the verifier. We prove that the complement $\bar{L}$ has a *constant-round* interactive proof of complexity that depends only exponentially on b. This provides the first evidence that for $\mathbf{NP}$-complete languages, we cannot expect interactive provers to be much more "laconic" than the standard $\mathbf{NP}$ proof.

When the proof system is further restricted (*e.g.,* when $b = 1$, or when we have perfect completeness), we get significantly better upper bounds on the complexity of $\bar{L}$.

[BGI$^+$01] On the (im)possibility of obfuscating programs (CRYPTO '01, with B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, and K. Yang) Informally, an *obfuscator* $\mathcal{O}$ is an (efficient, probabilistic) "compiler" that takes as input a program (or circuit) P and produces a new program $\mathcal{O}(P)$ that has the same functionality as P yet is "unintelligible" in some sense. Obfuscators, if they exist, would have a wide variety of cryptographic and complexity-theoretic applications, ranging from software protection to homomorphic encryption to complexity-theoretic analogues of Rice's theorem. Most of these applications are based on an interpretation of the "unintelligibility" condition in obfuscation as meaning that $\mathcal{O}(P)$ is a "virtual black box," in the sense that anything one can efficiently compute given $\mathcal{O}(P)$, one could also efficiently compute given oracle access to P.

In this work, we initiate a theoretical investigation of obfuscation. Our main result is that, even under very weak formalizations of the above intuition, obfuscation is impossible. We prove this by constructing a family of functions $\mathcal{F}$ that are *inherently unobfuscatable* in the following sense: there is a property $\pi : \mathcal{F} \to \{0,1\}$ such that (a) given *any program* that computes a function $f \in \mathcal{F}$, the value $\pi(f)$ can be efficiently computed, yet (b) given *oracle access* to a (randomly selected) function $f \in \mathcal{F}$, no efficient algorithm can compute $\pi(f)$ much better than random guessing.

5.2 Educational Accomplishments

I first discovered the rewards of teaching while an undergraduate at Harvard, where I was a teaching assistant for eight different courses in mathematics and computer science. Each of these courses involved teaching weekly sections, which I strove to make as engaging and useful to the students as possible. My hard work paid off in the satisfaction and success of my students, and I was awarded two *Certificates of Distinction in Teaching* based on the student evaluations of my teaching.

During the Summer 2000, I co-taught an intensive course entitled "Randomness and Computation" at the Summer Session on Computational Complexity Theory run by the Institute for Advanced Study and Park City Math Institute. The topics we covered were probabilistic proof systems and pseudorandomness. This proved to be an excellent opportunity to disseminate the exciting results in my areas of interest to a wider community, since the Summer Session attendees included high school teachers, undergraduates, graduate students, and active researchers in the mathematical sciences.

Although my teaching responsibilities at Harvard have not yet begun, I have already started working with a number of students. In addition to numerous informal discussions with students, I am supervising two theses for graduating seniors, I have supervised a sophomore doing an independent study in cryptography, and I have started working with my first Ph.D. student, Minh Nguyen.

I have also presented many papers at conferences such as FOCS, STOC, and CRYPTO, and I have also been given invited lectures at a number of universities, industrial research labs, and workshops. I will give an invited survey talk on pseudorandomness at the upcoming *RANDOM* conference in Berkeley, CA (Aug. 2001). Some workshops at which I have given invited presentations in the past include: *DIMACS Workshop on Randomization Methods in Algorithm Design* (Dec. 1997), *Fields Institute Workshop on Interactive Proofs, PCP's, and Fundamentals of Cryptography* (May 1998), *Oberwolfach Meeting on Complexity Theory* (Nov. 1998), *DIMACS Workshop on Pseudorandomness and Explicit Combinatorial Constructions* (Oct. 1999), *Oberwolfach Meeting on Complexity Theory* (Nov. 2000).

References

[AKS87] Miklós Ajtai, János Komlós, and E. Szemerédi. Deterministic simulation in LOGSPACE. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 132–140, New York City, 25–27 May 1987.

[AKS83] Miklós Ajtai, János Komlós, and Endre Szemerédi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3(1):1–19, 1983.

[AKL$^+$79] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, László Lovász, and Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science*, pages 218–223, San Juan, Puerto Rico, 29–31 October 1979. IEEE.

[AGM87] N. Alon, Z. Galil, and V. D. Milman. Better expanders and superconcentrators. *J. Algorithms*, 8(3):337–347, 1987.

[AM85] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Combin. Theory Ser. B*, 38(1):73–88, 1985.

[ALW01] Noga Alon, Alex Lubotzky, and Avi Wigderson. Semi-direct product in groups and zig-zag product in graphs. In *42nd Annual Symposium on Foundations of Computer Science*. IEEE, 14–17 October 2001. To appear.

[AS00] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, Inc., 2nd edition, 2000.

[ACR97] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. Worst-case hardness suffices for derandomization: A new method for hardness-randomness trade-offs. In Pierpaolo Degano, Robert Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Automata, Languages and Programming, 24th International Colloquium*, volume 1256 of *Lecture Notes in Computer Science*, pages 177–187, Bologna, Italy, 7–11 July 1997. Springer-Verlag.

[Arm98] Roy Armoni. On the derandomization of space-bounded computations. In M. Luby, J. Rolim, and M. Serna, editors, *Randomization and Approximation Techniques in Computer Science — RANDOM '98*, volume 1666 of *Lecture Notes in Computer Science*. Springer-Verlag, October 1998.

[ASWZ96] Roy Armoni, Michael Saks, Avi Wigderson, and Shiyu Zhou. Discrepancy sets and pseudorandom generators for combinatorial rectangles. In *37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996)*, pages 412–421. IEEE Comput. Soc. Press, Los Alamitos, CA, 1996.

[ATWZ00] Roy Armoni, Amnon Ta-Shma, Avi Wigderson, and Shiyu Zhou. An $O(\log(n)^{4/3})$ space algorithm for (s, t) connectivity in undirected graphs. *J. ACM*, 47(2):294–311, 2000.

[ALM96] Sanjeev Arora, F. T. Leighton, and Bruce M. Maggs. On-line algorithms for path selection in a nonblocking network. *SIAM J. Comput.*, 25(3):600–625, 1996.

[AS97] Sanjeev Arora and Madhu Sudan. Improved low degree testing and its applications. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 485–495, El Paso, Texas, 4–6 May 1997.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. *Comput. Complexity*, 1(1):3–40, 1991.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.

[BM88] László Babai and Shlomo Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[BNS89] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, pages 204–232, 15–17 May 1989.

[BGW99] Ziv Bar-Yossef, Oded Goldreich, and Avi Wigderson. Deterministic amplification of space-bounded probabilistic algorithms. In *Proceedings of the Fourteenth Annual Conference on Computational Complexity*, pages 188–198. IEEE, May 4–6 1999.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagaliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *Advances in Cryptology—CRYPTO '01*, Lecture Notes in Computer Science. Springer-Verlag, 2001, August 2001. To appear.

[BF90] Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *7th Annual Symposium on Theoretical Aspects of Computer Science*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48, Rouen, France, 22–24 February 1990. Springer.

[BHSV98] Mihir Bellare, Shai Halevi, Amit Sahai, and Salil Vadhan. Many-to-one trapdoor functions and their relation to public-key cryptosystems. In H. Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, number 1462 in Lecture Notes in Computer Science. Springer, 1998.

[BFR$^+$98] Michael Bender, Antonio Fernández, Dana Ron, Amit Sahai, and Salil Vadhan. The power of a pebble: Exploring and mapping directed graphs. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 269–278, Dallas, TX, May 1998. ACM. Full version to appear in *Information and Computation*.

[Blo70] Burton H. Bloom. Space/time tradeoffs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.

[BG84] Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology: Proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 289–299. Springer-Verlag, 1985, 19–22 August 1984.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, November 1984.

[BMRS00] Harry Buhrman, Peter Bro Miltersen, Jaikumar Radhakrishnan, and Venkatesh Srinivasan. Are bitvectors optimal? In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, pages 449–458, 21–23 May 2000.

[CDH$^+$00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In Bart Preneel, editor, *Advances in Cryptology—EUROCRYPT 00*, volume 1807 of *Lecture Notes in Computer Science*, pages 453–469. Springer-Verlag, 24–28 May 2000.

[FKS84] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31(3):538–544, 1984.

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, June 1981.

[Gol99] Oded Goldreich. *Modern cryptography, probabilistic proofs and pseudorandomness*. Springer-Verlag, Berlin, 1999.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[GH98] Oded Goldreich and Johan Håstad. On the complexity of interactive proofs with bounded communication. *Information Processing Letters*, 67(4):205–214, 1998.

[GIL$^+$90] Oded Goldreich, Russell Impagliazzo, Leonid Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *31st Annual Symposium on Foundations of Computer Science*, volume I, pages 318–326, St. Louis, Missouri, 22–24 October 1990. IEEE.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991.

[GSV98] Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 399–408, Dallas, TX, May 1998. ACM.

[GSV99] Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero-knowledge be made non-interactive?, or On the relationship of SZK and NISZK. In M. Wiener, editor, *Advances in Cryptology—CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 467–484. Springer-Verlag, 1999, 15–19 August 1999.

[GV99] Oded Goldreich and Salil Vadhan. Comparing entropies in statistical zero-knowledge with applications to the structure of SZK. In *Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, pages 54–73, Atlanta, GA, May 1999. IEEE Computer Society Press.

[GVW00] Oded Goldreich, Salil Vadhan, and Avi Wigderson. Simplified derandomization of BPP using a hitting set generator. Technical Report TR00-04, Electronic Colloquium on Computational Complexity, January 2000.

[GVW01] Oded Goldreich, Salil Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. In *Automata, Languages and Programming, 28th International Colloquium*, Lecture Notes in Computer Science, Crete, Greece, 7–11 July 2001. Springer-Verlag.

[GZ97] Oded Goldreich and David Zuckerman. Another proof that BPP $\subseteq$ PH (and more). *Electronic Colloquium on Computational Complexity* Technical Report TR97-045, September 1997. `http://www.eccc.uni-trier.de/eccc`.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.

[GS89] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In Silvio Micali, editor, *Advances in Computing Research*, volume 5, pages 73–90. JAC Press, Inc., 1989.

[Gro00] Misha Gromov. Spaces and questions. *Geometric and Functional Analysis*, pages 118–161, 2000. Part I of Special Volume on GAFA 2000 (Tel Aviv, 1999).

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396 (electronic), 1999.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science*, pages 538–545, Milwaukee, Wisconsin, 23–25 October 1995. IEEE.

[IKW01] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. In *Proceedings of the Sixteenth Annual Conference on Computational Complexity*. IEEE, June 18–21 2001.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 356–364, Montréal, Québec, Canada, 23–25 May 1994.

[ISW00] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, pages 1–10, Portland, Oregon, May 2000. See also ECCC TR00-009.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, El Paso, Texas, 4–6 May 1997.

[IW98] Russell Impagliazzo and Avi Wigderson. Randomness vs. time: De-randomization under a uniform assumption. In *36th Annual Symposium on Foundations of Computer Science*, Palo Alto, CA, November 8–11 1998. IEEE.

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *Society for Industrial and Applied Mathematics Journal on Computing*, 18(6):1149–1178, December 1989.

[JSV01] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, pages 712–721, Crete, Greece, 6–8 July 2001.

[JM87] Shuji Jimbo and Akira Maruoka. Expanders obtained from affine transformations. *Combinatorica*, 7(4):343–355, 1987.

[Kah95] Nabil Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM*, 42(5):1091–1106, September 1995.

[KL82] Richard M. Karp and Richard J. Lipton. Turing machines that take advice. *L'Enseignement Mathématique. Revue Internationale. IIe Série*, 28(3-4):191–209, 1982.

[LLSZ97] Nathan Linial, Michael Luby, Michael Saks, and David Zuckerman. Efficient construction of a small hitting set for combinatorial rectangles in high dimension. *Combinatorica*, 17(2):215–234, 1997.

[Lip89] Richard Lipton. New directions in testing. In *Proceedings of DIMACS Workshop on Distributed Computing and Cryptography*, 1989.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[LW93] A. Lubotzky and B. Weiss. Groups and expanders. In *Expanding graphs (Princeton, NJ, 1992)*, pages 95–109. Amer. Math. Soc., Providence, RI, 1993.

[LP01] Alexander Lubotzky and Igor Pak. The product replacement algorithm and Kazhdan's property (T). *Journal of the American Mathematical Society*, 14(2):347–363 (electronic), 2001.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, October 1992.

[Mar73] G. A. Margulis. Explicit constructions of expanders. *Problemy Peredǎci Informacii*, 9(4):71–80, 1973.

[Mar88] G. A. Margulis. Explicit group-theoretic constructions of combinatorial schemes and their applications in the construction of expanders and concentrators. *Problemy Peredachi Informatsii*, 24(1):51–60, 1988.

[MW00] Ueli Maurer and Stefan Wolf. Information-theoretic key agreement: From weak to strong secrecy for free. In Bart Preneel, editor, *Advances in Cryptology—EUROCRYPT 00*, volume 1807 of *Lecture Notes in Computer Science*. Springer-Verlag, 24–28 May 2000.

[MRV99] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science*, New York, NY, October 1999. IEEE.

[Mil76] Gary L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13(3):300–317, 1976. Working papers presented at the ACM-SIGACT Symposium on the Theory of Computing (Albuquerque, N.M., 1975).

[Mit01] Michael Mitzenmacher. Compressed bloom filters. In *Proceedings of the 20th Annual Symposium on Principles of Distributed Computing*. ACM, August 26–29 2001. To appear.

[Mor94] Moshe Morgenstern. Existence and explicit constructions of $q+1$ regular Ramanujan graphs for every prime power q. *J. Combin. Theory Ser. B*, 62(1):44–62, 1994.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, August 1993.

[Nil91] A. Nilli. On the second eigenvalue of a graph. *Discrete Math.*, 91(2):207–210, 1991.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, October 1994.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, February 1996.

[Pin73] M. Pinsker. On the complexity of a concentrator. In *7th Annual Teletraffic Conference*, pages 318/1–318/4, Stockholm, 1973.

[PY82] N. Pippenger and A.C. Yao. Rearrangeable networks with limited depth. *SIAM J. Algebraic and Discrete Methods*, 3:411–417, 1982.

[Pip87] Nicholas Pippenger. Sorting and selecting in rounds. *SIAM Journal on Computing*, 16(6):1032–1038, December 1987.

[Rab80] Michael O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12(1):128–138, 1980.

[RR99] Ran Raz and Omer Reingold. On recycling the randomness of the states in space bounded computation. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, Atlanta, GA, May 1999.

[RRV99a] Ran Raz, Omer Reingold, and Salil Vadhan. Error reduction for extractors. In *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science*, New York, NY, October 1999. IEEE.

[RRV99b] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 149–158, Atlanta, GA, May 1999. ACM. Invited to special issue of *Journal of Computer and System Sciences*.

[RR97] Alexander Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[RSW00] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. In *41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, CA, 17–19 October 2000. IEEE.

[RVW00] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 3–13, Redondo Beach, CA, 17–19 October 2000. IEEE. Journal version accepted to *Annals of Mathematics* (subject to minor revisions).

[RZ98] Alexander Russell and David Zuckerman. Perfect information leader election in $\log^* n + o(1)$ rounds. In *39th Annual Symposium on Foundations of Computer Science*, Palo Alto, California, 8–11 November 1998. IEEE.

[SV97] Amit Sahai and Salil Vadhan. A complete promise problem for statistical zero-knowledge. In *Proceedings of the 38th Annual Symposium on the Foundations of Computer Science*, pages 448–457, Miami Beach, FL, October 1997. IEEE.

[SV99] Amit Sahai and Salil Vadhan. Manipulating statistical difference. In Panos Pardalos, Sanguthevar Rajasekaran, and José Rolim, editors, *Randomization Methods in Algorithm Design (DIMACS Workshop, December 1997)*, volume 43 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 251–270. American Mathematical Society, 1999.

[SZ99] Michael Saks and Shiyu Zhou. $\mathrm{bp_h space}(S) \subseteq \mathrm{dspace}(S^{3/2})$. *Journal of Computer and System Sciences*, 58(2):376–403, 1999. 36th IEEE Symposium on the Foundations of Computer Science (Milwaukee, WI, 1995).

[Sha01] Ronen Shaltiel. Towards proving strong direct product theorems. In *Proceedings of the Sixteenth Annual Conference on Computational Complexity*. IEEE, June 18–21 2001.

[SU01] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudo-random generator. In *42nd Annual Symposium on Foundations of Computer Science*. IEEE, 14–17 October 2001. To appear.

[Sha92] Adi Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, October 1992.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.

[Sip88] Michael Sipser. Expanders, randomness, or time versus space. *Journal of Computer and System Sciences*, 36(3):379–383, June 1988.

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1710–1722, 1996. Codes and complexity.

[SS77] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6(1):84–85, 1977.

[Spi96] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1723–1731, 1996. Codes and complexity.

[Sud97] Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[Sud00] Madhu Sudan. List decoding: Algorithms and applications. *SIGACT News*, 31(1):16–27, March 2000.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62:236–266, 2001. Special issue on CCC '99. Extended abstract in *STOC–CCC '99* joint session.

[TUZ01] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Loss-less condensers, unbalanced expanders, and extractors. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, pages 143–162, Crete, Greece, 6–8 July 2001.

[TZ01] Amnon Ta-Shma and David Zuckerman. Extractor codes. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, pages 193–199, Crete, Greece, 6–8 July 2001.

[TZS01] Amnon Ta-Shma, David Zuckerman, and Shmuel Safra. Extractors from Reed–Muller codes. In *42nd Annual Symposium on Foundations of Computer Science*. IEEE, 14–17 October 2001. To appear.

[Tan84] Michael R. Tanner. Explicit concentrators from generalized n-gons. *SIAM Journal on Algebraic Discrete Methods*, 5(3):287–293, 1984.

[Tan81] R. Michael Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981.

[Tod91] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[Tre99] Luca Trevisan. Construction of extractors using pseudo-random generators. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 141–148, Atlanta, GA, May 1999. See also ECCC TR98-55.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, CA, 17–19 October 2000. IEEE.

[Uma99] Christopher Umans. Hardness of approximating Σ_2^p minimization problems. In *40th Annual Symposium on Foundations of Computer Science*, pages 465–474, New York, NY, 17–19 October 1999. IEEE.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *J. Assoc. Comput. Mach.*, 34(1):209–219, 1987.

[Vad00] Salil Vadhan. Lecture notes on interactive proofs & zero-knowledge proofs. In *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*. American Mathematical Society, 2000. To appear.

[Vad95] Salil P. Vadhan. *The Complexity of Counting*. Undergraduate thesis, Harvard University, Cambridge, MA, 1995. Won *Hoopes Prize* for outstanding undergraduate work at Harvard University.

[Vad97] Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs, May 1997. To appear in *SIAM Journal on Computing*.

[Vad99] Salil P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, Massachusetts Institute of Technology, August 1999. To be published by Springer-Verlag for winning the *ACM Doctoral Dissertation Award 2000*.

[Vad00] Salil P. Vadhan. On transformations of interactive proofs that preserve the prover's complexity. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, Portland, OR, May 2000. ACM.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Mathematical foundations of computer science (Proc. Sixth Sympos., Tatranská Lomnica, 1977)*, pages 162–176. Lecture Notes in Comput. Sci., Vol. 53. Springer, Berlin, 1977.

[Val79] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.

[WZ99] Avi Wigderson and David Zuckerman. Expanders that beat the eigenvalue bound: explicit construction and applications. *Combinatorica*, 19(1):125–138, 1999.

[Yao82] Andrew C. Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, 3–5 November 1982. IEEE.

[Zuc96] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, October/November 1996.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997.