

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Venkatesan Guruswami

List Decoding of Error-Correcting Codes

Winning Thesis of the
2002 ACM Doctoral Dissertation Competition

Author

Venkatesan Guruswami
University of Washington
Department of Computer Science and Engineering
Seattle, WA 98195-2350, USA
E-mail: venkat@cs.washington.edu

Library of Congress Control Number: 2004115727

CR Subject Classification (1998): E.4, F.2.2, H.1.1, G.2.1

ISSN 0302-9743

ISBN 3-540-24051-9 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN: 11365877 06/3142 5 4 3 2 1 0

To my parents

Foreword

How can one exchange information effectively when the medium of communication introduces errors? This question has been investigated extensively starting with the seminal works of Shannon (1948) and Hamming (1950), and has led to the rich theory of “error-correcting codes”. This theory has traditionally gone hand in hand with the algorithmic theory of “decoding” that tackles the problem of recovering from the errors *efficiently*. This thesis presents some spectacular new results in the area of decoding algorithms for error-correcting codes. Specifically, it shows how the notion of “list-decoding” can be applied to recover from far more errors, for a wide variety of error-correcting codes, than achievable before.

A brief bit of background: *error-correcting codes* are combinatorial structures that show how to represent (or “encode”) information so that it is resilient to a moderate number of errors. Specifically, an error-correcting code takes a short binary string, called the message, and shows how to transform it into a longer binary string, called the codeword, so that if a small number of bits of the codeword are flipped, the resulting string does not look like any other codeword. The maximum number of errors that the code is guaranteed to detect, denoted d , is a central parameter in its design. A basic property of such a code is that if the number of errors that occur is known to be smaller than $d/2$, the message is determined uniquely. This poses a computational problem, called the decoding problem: compute the message from a corrupted codeword, when the number of errors is less than $d/2$. While naive decoding algorithms run in time exponential in d , sophisticated algorithms with polynomial running time have been found for a variety of codes, enabling widespread usage of error-correcting codes.

The principal concern of this thesis is the question: “What happens when the number of errors that occur is more than $d/2$?” This question is important for practical purposes, so that one can extract more out of any given communication channel. Furthermore, the central nature of error-correcting codes in the theory of computer science makes this question an important one in this domain as well. It is well known that if the number of errors exceed $d/2$, then the message may potentially not be recoverable uniquely. However, it is conceivable that one can pin down a small list of candidate messages that include the intended message. This possibility motivated Elias

(1957) and Wozencraft (1958) to define the list-decoding problem: “Given a corrupted codeword and an error parameter e , compute a list of all codewords that differ from the corrupted word in most e places.”

Even though the list-decoding problem had been in existence for several decades, it did not meet with algorithmic success till 1997. In the last ten years or so, however, this area has seen some remarkable advances, and these results represent the original contributions of this thesis. List-decoding algorithms are presented for a wide variety of codes considered in the literature including “Reed-Solomon codes”, “algebraic-geometry codes”, “concatenated codes”, and “graph-theoretic codes”. In addition to describing new results, the thesis also serves as a valuable source of reference on list-decoding. It introduces the topic gently, re-examining the definition, explaining why it is interesting and then describing the central combinatorial and algorithmic problems in this domain. It includes a nice survey of prior combinatorial work most of which is scattered in the literature. After covering the new algorithmic results, the thesis includes an excellent survey of the many applications of list-decoding in theoretical computer science including “hardness amplification”, “extracting randomness”, and “pseudorandomness”.

The style of the exposition is crisp and the enormous amount of information is presented in a clear, structured form. This thesis will be valuable to readers interested in mathematical aspects of computer science or communication.

August 2004

Madhu Sudan
Professor of Computer Science
MIT, Cambridge, MA, USA.

Preface

Error-correcting codes are combinatorial objects designed to cope with the problem of reliable transmission of information on a noisy channel. A fundamental algorithmic challenge in coding theory and practice is to efficiently decode the original transmitted message even when a few symbols of the received word are in error. The naive search algorithm runs in exponential time, and several classical polynomial time decoding algorithms are known for specific code families. Traditionally, however, these algorithms have been constrained to output a unique codeword. Thus they faced a “combinatorial barrier” and could only correct up to $d/2$ errors, where d is the minimum distance of the code.

An alternate notion of decoding called *list decoding*, proposed independently by Elias and Wozencraft in the late 1950s, allows the decoder to output a list of *all* codewords that differ from the received word in a certain number of positions. Even when constrained to output a relatively small number of answers, list decoding permits recovery from errors well beyond the $d/2$ barrier, and opens up the possibility of meaningful error correction from large amounts of noise. However, for nearly four decades after its conception, this potential of list decoding was largely untapped due to the lack of *efficient* algorithms to list decode beyond $d/2$ errors for useful families of codes.

This book presents a detailed investigation of list decoding, and proves its potential, feasibility, and importance as a combinatorial and algorithmic concept. The results discussed in the book are divided into three parts: the first one on combinatorial results, the second on polynomial time list decoding algorithms, and the third on applications. We describe each of the parts in further detail below.

Part I deals with the combinatorics of list decoding and attempts to sharpen our understanding of the potential and limits of list decoding, and its relation to more classical coding-theoretic parameters like the rate and minimum distance. A combinatorial bound called the Johnson bound asserts that codes with large minimum distance have a large list decoding radius, and this raises algorithmic questions on list decoding such codes from a large number of errors for central codes that are known to have good distance properties. This is not the only approach to obtaining good list decodable codes, and in fact directly optimizing the list decoding radius leads to better trade-offs as

a function of the rate of the code (as can be shown by applications of the probabilistic method). Part I can be summed up with the statement: *good codes with excellent combinatorial list decodability properties exist*. This sets the stage for the algorithmic results of Part II by highlighting what one can and cannot hope to do with list decoding, and poses the challenge of tapping the potential of list decoding with efficient algorithms.

Part II comprises the crux of the book, namely its algorithmic results, which were lacking in the early works on list decoding. The algorithmic results attempt to “match” the combinatorial bounds with explicit code constructions and efficient decoding algorithms. Our algorithmic results include:

- Efficient list decoding algorithms for classically studied codes such as Reed-Solomon codes and algebraic-geometric codes. In particular, building upon an earlier algorithm by Sudan, we present the *first* polynomial time algorithm to decode Reed-Solomon codes beyond $d/2$ errors for every value of the rate.
- A new *soft* list decoding algorithm for Reed-Solomon and algebraic-geometric codes, and novel decoding algorithms for concatenated codes based on it.
- New code constructions using concatenation and/or expander graphs that have good (and sometimes near-optimal) rates and are efficiently list decodable from extremely large amounts of noise.
- Error-correcting codes with good (and sometimes near-optimal rates) for list decoding from erasures.

Part II can be summed up with the statement: *there exist “explicit” constructions of “good” codes together with efficient list decoding algorithms*.

In Part III, we discuss some applications of the results and techniques from earlier chapters to domains both within and outside of coding theory. Using an expander-based construction in the same spirit as our construction for list decoding, we get a significant improvement over a prior result for *unique decoding*. Specifically, we construct *linear time* encodable and decodable codes that match the trade-off between rate and error-correction radius achieved by the best known constructions with polynomial time decoding (and in fact the trade-off is almost the best possible over large alphabets). This constitutes a vast improvement compared with previous constructions of linear time codes that could only correct a tiny fraction of errors with positive rates. The notion of list decoding turns out to be central to certain contexts in theoretical computer science outside of coding theory, for example in complexity theory, cryptography, and algorithms. For these applications unique decoding does not suffice, and moreover, for several of them one needs *efficient* list decoding algorithms.

A detailed chapter by chapter description of the contents can be found in Section 2.3.

Acknowledgments

We know too much for one man to know much.

J. Robert Oppenheimer

This monograph is a revised version of my doctoral dissertation, written under the supervision of Madhu Sudan and submitted to MIT in August 2001. I am grateful to MIT for nominating my Ph.D. thesis for the ACM Doctoral Dissertation Award competition, and to ACM and the awards committee for awarding the honor to my dissertation.

My first and foremost acknowledgment is to my advisor Madhu Sudan. When I made a decision to go to MIT for grad school in the spring of 1997, I was not aware that Madhu Sudan would be joining its faculty that Fall, so it was quite serendipitous that I got him as my advisor. While I found MIT to be every bit the wonderful place I had anticipated it to be and more, Madhu was the most important reason my academic experience at MIT was so enjoyable and fulfilling. For the wonderful collaboration which led to several of the key chapters of my thesis, for all his patient advice, help and support on matters technical and otherwise, and for all the things I learned from him during my stay at MIT and continue to do so, I will be forever grateful to Madhu.

I am most grateful to Madhu Sudan, Johan Håstad, Piotr Indyk, Amit Sahai, and David Zuckerman for their collaboration which led to several of the results discussed in this monograph. Collectively, this is as much, if not more, their book as it is mine. I also wish to thank the several other people with whom I have had useful discussions on coding theory and related topics. These include Noga Alon, Sanjeev Arora, Sasha Barg, Moses Charikar, Yevgeniy Dodis, Peter Elias, Sanjeev Khanna, Subhash Khot, Ralf Koetter, Ravi Kumar, Hendrik Lenstra, Daniele Micciancio, Jaikumar Radhakrishnan, Amin Shokrollahi, D. Sivakumar, Dan Spielman, Luca Trevisan, Salil Vadhan, and Alex Vardy, though undoubtedly I have left out several others.

A special thanks is due to the members of my thesis reading committee at MIT: Peter Elias, Dan Spielman, and Madhu Sudan. Technically, it was only appropriate that I had these three people on my committee: Peter first defined the notion of list decoding; Madhu discovered the first non-trivial efficient list decoding algorithm; and Dan constructed the first linear-time codes (the subject of Chapter 11 of this book). I regret that I will not be able to present a personal copy of the book to Peter, who sadly left us a few months after I submitted my thesis to MIT.

It is with really fond memories that I acknowledge the stimulating working atmosphere and the company of a great group of friends and colleagues that I found in MIT's theory group. The good time I had at MIT owes a lot to the wonderful student body I had the privilege of being a part of. I would like to thank Salil, Yevgeniy, Eric, Amit, Raj, Anna, Sofya, Adam K., Adam S., Maria, Matthias, Feldman, Abhi, Rocco, Daniele, Alantha, Ryan, Prahladh,

and many others, for numerous conversations on all sorts of topics, and for making my life at MIT LCS so much fun. I was lucky that Luca Trevisan was at MIT the year I started; from him I learned a lot, and with him (and Danny Lewin and Madhu) I shared my first research experience in graduate school. In my last year at MIT I benefited immensely from the time I spent working and hanging out with Piotr Indyk, for which I sincerely thank him. I relish very much our continuing collaboration on expander codes. Lars Engebretsen, the other member of our espresso trio, also contributed greatly to making my final year at MIT so memorable.

My sincere thanks to the theory group staff, and in particular Joanne Talbot and Be Blackburn, for their good cheer and all their administrative and other help.

It is a pleasure to acknowledge my current academic home, University of Washington CSE, for its warm and congenial atmosphere, with special thanks to my theory colleagues Paul Beame, Anna Karlin and Richard Ladner for their support and company.

A *huge* thanks to all my friends whom I met at various junctures of my life. True friends are those who take pride in your achievements, and I am grateful that I have several who meet this definition and who are an inseparable part of my life.

I owe a lot to two professors from college: C. Pandu Rangan for encouraging me in every possible way and getting me started on research well before I started grad school; and S. A. Choudum whose wonderful Graph Theory course sparked my interest in algorithmic graph theory and eventually theoretical computer science.

My most important acknowledgment is to my close and loving family: my parents and my sister Shantha, who have filled my life with joy and who mean the world to me. Many thanks to Vaishnavi, my most fortunate discovery, for her cheer and providing useful distractions during the course of this revision.

Words cannot express my thanks to my parents for all that they have gone through and done for me. So, of all the sentences in this book none was easier to write than this one: To my parents, this book is dedicated with love.

Seattle, Washington
August 2004

Venkatesan Guruswami

I gratefully acknowledge the fellowships and grants that supported my research at MIT. My research was supported in part by funding from NSF CCR 9875511, NSF CCR 9912342, and NTT Award MIT 2001-04, and in part by an IBM Graduate Fellowship.

Contents

1	Introduction	1
1.1	Basics of Error-Correcting Codes	1
1.2	The Decoding Problem for Error-Correcting Codes	4
1.3	List Decoding	7
1.3.1	Definition	7
1.3.2	Is List Decoding a Useful Relaxation of Unique Decoding?	8
1.3.3	The Challenge of List Decoding	9
1.3.4	Early Work on List Decoding	10
1.4	Contributions of This Work	10
1.5	Background Assumed of the Reader	13
1.6	Comparison with Doctoral Thesis Submitted to MIT	13
2	Preliminaries and Monograph Structure	15
2.1	Preliminaries and Definitions	15
2.1.1	Basic Definitions for Codes	15
2.1.2	Code Families	17
2.1.3	Linear Codes	17
2.1.4	Definitions Relating to List Decoding	18
2.1.5	Commonly Used Notation	20
2.2	Basic Code Families	20
2.2.1	Reed-Solomon Codes	20
2.2.2	Reed-Muller Codes	21
2.2.3	Algebraic-Geometric Codes	22
2.2.4	Concatenated Codes	22
2.2.5	Number-Theoretic Codes	23
2.3	Detailed Description of Book Chapters	24
2.3.1	Combinatorial Results	24
2.3.2	Algorithmic Results	26
2.3.3	Applications	28
2.3.4	Conclusions	29
2.3.5	Dependencies Among Chapters	29

Part I Combinatorial Bounds

3	Johnson-Type Bounds and Applications to List Decoding .	33
3.1	Introduction	33
3.2	Definitions and Notation	34
3.3	The Johnson Bound on List Decoding Radius	35
3.3.1	Proof of Theorem 3.1	37
3.3.2	Geometric Lemmas	39
3.4	Generalization in Presence of Weights	41
3.5	Notes	43
4	Limits to List Decodability	45
4.1	Introduction	45
4.2	Informal Description of Results	46
4.3	Formal Description of Results	47
4.3.1	The Result for Non-linear Codes	47
4.3.2	Definitions	47
4.3.3	Statement of Results	48
4.4	Super-constant List Size at Johnson Radius	50
4.4.1	The Basic Construction	50
4.4.2	Related Constructions	55
4.4.3	The Technical “Linear-Algebraic” Lemma	56
4.5	Super-polynomial List Size Below Minimum Distance	60
4.5.1	Proof of Theorem 4.10	60
4.6	Explicit Constructions with Polynomial-Sized Lists	62
4.6.1	Fourier Analysis and Group Characters	62
4.6.2	Idea Behind the Construction	63
4.6.3	Proof of Theorem 4.8	64
4.6.4	Proof of Theorem 4.9	67
4.6.5	Proof of Theorem 4.16	68
4.7	Super-polynomial List Sizes at the Johnson Bound	71
4.7.1	Proof Idea	71
4.7.2	The Technical Proof	72
4.7.3	Unconditional Proof of Tightness of Johnson Bound	76
4.8	Notes and Open Questions	76
5	List Decodability Vs. Rate	79
5.1	Introduction	79
5.2	Definitions	80
5.3	Main Results	81
5.3.1	Basic Lower Bounds	81
5.3.2	An Improved Lower Bound for Binary Linear Codes	85
5.3.3	Upper Bounds on the Rate Function	88
5.3.4	“Optimality” of Theorem 5.8	89

5.4	Prelude to Pseudolinear Codes	90
5.5	Notes	91

Part II Code Constructions and Algorithms

6	Reed-Solomon and Algebraic-Geometric Codes	95
6.1	Introduction	95
6.1.1	Reed-Solomon Codes	96
6.1.2	Algebraic-Geometric Codes	97
6.1.3	Soft-Decision Decoding Algorithms	98
6.2	Reed-Solomon Codes	98
6.2.1	Reformulation of the Problem	98
6.2.2	Informal Description of the Algorithm	100
6.2.3	Formal Description of the Algorithm	102
6.2.4	Correctness of the Algorithm	103
6.2.5	A “Geometric” Example	105
6.2.6	Results for Specific List Sizes	108
6.2.7	Runtime of the Algorithm	111
6.2.8	Main Theorems About Reed-Solomon List Decoding ..	113
6.2.9	Some Further Consequences	114
6.2.10	Weighted Polynomial Reconstruction and Soft Decoding of RS Codes	117
6.3	Algebraic-Geometric Codes	121
6.3.1	Overview	121
6.3.2	Algebraic-Geometric Codes: Preliminaries	122
6.3.3	List Decoding Algorithm for Algebraic-Geometric Codes	126
6.3.4	Root Finding over Algebraic Function Fields	129
6.3.5	An Explicit List Decoding Algorithm	132
6.3.6	Analysis of the Algorithm	134
6.3.7	Weighted List Decoding of AG-codes	136
6.3.8	Decoding Up to the “ q -ary Johnson Radius”	137
6.3.9	List Decodability Offered by the Best-Known AG-codes	138
6.4	Concluding Remarks and Open Questions	141
6.5	Bibliographic Notes	142
7	A Unified Framework for List Decoding of Algebraic Codes	147
7.1	Introduction	147
7.1.1	Overview	148
7.2	Preliminaries	149
7.3	Ideal-Based Codes	151
7.3.1	Examples of Ideal-Based Codes	151

7.4	Properties of Ideal-Based Codes	152
7.4.1	Axioms and Assumptions	152
7.4.2	Distance Property of Ideal-Based Codes	153
7.5	List Decoding Ideal-Based Codes	153
7.5.1	High Level Structure of the Decoding Algorithm	154
7.5.2	Formal Description of the Decoding Algorithm	155
7.5.3	Further Assumptions on the Underlying Ring and Ideals	156
7.5.4	Analysis of the List Decoding Algorithm	156
7.5.5	Performance of the List Decoding Algorithm	159
7.5.6	Obtaining Algorithms for Reed-Solomon and AG-codes	160
7.6	Decoding Algorithms for CRT Codes	161
7.6.1	Combinatorial Bounds on List Decoding	162
7.6.2	Weighted List Decoding Algorithm	165
7.6.3	Applications to “Interesting” Weight Settings	169
7.7	GMD Decoding for CRT Codes	171
7.8	Bibliographic Notes	174
8	List Decoding of Concatenated Codes	177
8.1	Introduction	177
8.2	Context and Motivation of Results	178
8.3	Overview of Results	179
8.4	Decoding Concatenated Codes with Inner Hadamard Code	180
8.4.1	Reed-Solomon Concatenated with Hadamard Code	184
8.4.2	AG-code Concatenated with Hadamard Code	186
8.4.3	Consequence for Highly List Decodable Codes	187
8.5	Decoding a General Concatenated Code with Outer Reed-Solomon or AG-code	187
8.5.1	A Relevant Combinatorial Result	188
8.5.2	The Formal Decoding Algorithm and Its Analysis	192
8.5.3	Consequence for Highly List Decodable Codes	195
8.6	Improved Rate Using Tailor-Made Concatenated Code	198
8.6.1	The Inner Code Construction	199
8.6.2	The Concatenated Code and the Decoding Algorithm	202
8.7	Open Questions	205
8.8	Bibliographic Notes	206
9	New, Expander-Based List Decodable Codes	209
9.1	Introduction	209
9.2	Overview of Results and Techniques	210
9.2.1	Main Results	210
9.2.2	Our Techniques	213
9.2.3	A Useful Definition	215

9.3	Pseudolinear Codes: Existence Results and Properties	215
9.3.1	Pseudolinear (Code) Families	216
9.3.2	Probabilistic Constructions of Good, List Decodable Pseudolinear Codes	218
9.3.3	Derandomizing Constructions of Pseudolinear Codes	221
9.3.4	Faster Decoding of Pseudolinear Codes over Large Alphabets	225
9.4	The Basic Expander-Based Construction of List Decodable Codes	227
9.4.1	Definition of Required “Expanders”	228
9.4.2	Reduction of List Decoding to List Recoverability Using Dispersers	228
9.4.3	Codes of Rate $\Omega(\varepsilon^2)$ List Decodable to a Fraction ($1 - \varepsilon$) of Errors	231
9.4.4	Better Rate with Sub-exponential Decoding	234
9.5	Constructions with Better Rate Using Multi-concatenated Codes	235
9.5.1	The Basic Multi-concatenated Code	236
9.5.2	Codes of Rate $\Omega(\varepsilon)$ with Sub-exponential List Decoding for a Fraction ($1 - \varepsilon$) of Errors	239
9.5.3	Binary Codes of Rate $\Omega(\varepsilon^3)$ with Sub-exponential List Decoding Up to a Fraction ($1/2 - \varepsilon$) of Errors	242
9.6	Improving the Alphabet Size: Juxtaposed Codes	243
9.6.1	Intuition	244
9.6.2	The Actual Construction	245
9.7	Notes	249
10	List Decoding from Erasures	251
10.1	Introduction	251
10.2	Overview	252
10.3	Definitions	253
10.3.1	Comment on Combinatorial Vs. Algorithmic Erasure List-Decodability	254
10.4	Relation to Generalized Hamming Weights	254
10.5	Erasure List-Decodability and Minimum Distance	256
10.6	Combinatorial Bounds for Erasure List-Decodability	257
10.6.1	Discussion	257
10.6.2	Lower Bound on $\tilde{R}_L(p)$	258
10.6.3	Lower Bound on $\tilde{R}_L^{\text{lin}}(p)$	259
10.6.4	Upper Bound on $\tilde{R}_L(p)$	262
10.6.5	Improved Upper Bound for $\tilde{R}_L^{\text{lin}}(p)$	265
10.6.6	Provable Separation Between Erasure List-Decodable Linear and Non-linear Codes	265

10.7	A Good Erasure List-Decodable Binary Code Construction ..	265
10.7.1	Context	265
10.7.2	The Formal Result	266
10.7.3	Obtaining Near-Linear Encoding and Decoding Times	268
10.7.4	The ε^2 “Rate Barrier” for Binary Linear Codes	270
10.8	Better Results for Larger Alphabets Using Juxtaposed Codes	272
10.8.1	Main Theorem	272
10.8.2	Improving the Decoding Time in Theorem 10.22	275
10.9	Concluding Remarks	276
10.10	Bibliographic Notes	277

Part III Applications

11	Linear-Time Codes for Unique Decoding	283
11.1	Context and Introduction	283
11.2	Background on Expanders	284
11.3	Linear-Time Encodable and Decodable Codes: Construction I	285
11.3.1	Codes with Rate $\Omega(\varepsilon^2)$ Decodable Up to a Fraction ($1/2 - \varepsilon$) of Errors	286
11.3.2	Binary Codes with Rate $\Omega(\varepsilon^4)$ Decodable Up to a Fraction ($1/4 - \varepsilon$) of Errors	288
11.4	Linear-Time Codes with Near-Optimal Rate	289
11.4.1	High-Level View of the Construction	289
11.4.2	Linear-Time Codes with Rates Close to 1	291
11.4.3	Linear-Time Error-Correcting Codes Meeting the Singleton Bound	294
11.5	Linear-Time Encodable Binary Codes Meeting the Zyablov Bound	297
11.6	Bibliographic Notes	298
12	Sample Applications Outside Coding Theory	299
12.1	An Algorithmic Application: Guessing Secrets	299
12.1.1	Formal Problem Description	300
12.1.2	An Explicit Strategy with $O(\log N)$ Questions	302
12.1.3	An Efficient Algorithm to Recover the Secrets	304
12.1.4	The Case of More than Two Secrets	308
12.1.5	An Efficient “Partial Solution” for the k -Secrets Game	309
12.2	Applications to Complexity Theory	310
12.2.1	Hardcore Predicates from One-Way Permutations	311
12.2.2	Hardness Amplification of Boolean Functions	313

12.2.3	Average-Case Hardness of Permanent	315
12.2.4	Extractors and Pseudorandom Generators	315
12.2.5	Membership Comparable Sets	318
12.2.6	Inapproximability of NP Witnesses	320
12.3	Applications to Cryptography	324
12.3.1	Cryptanalysis of Certain Block Ciphers	324
12.3.2	Finding Smooth Integers	325
12.3.3	Efficient Traitor Tracing	325
13	Concluding Remarks	329
13.1	Summary of Contributions	329
13.2	Directions for Future Work	330
13.2.1	Some Specific Open Questions	330
13.2.2	Construction of “Capacity-Approaching” List Decodable Codes	331
A	GMD Decoding of Concatenated Codes	333
A.1	Proof	333
	References	337
	Index	349