Jörg Rothe

Complexity Theory and Cryptology

An Introduction to Cryptocomplexity

SPIN Springer's internal project number, if known

– Monograph –

January 10, 2005

Springer
Berlin Heidelberg New York
Barcelona Hong Kong
London Milan Paris
Tokyo

To Irene, Paula, and Ella

Preface

This book is an accessible introduction to complexity theory and cryptology, two closely related areas in theoretical computer science. Based on courses taught at Heinrich-Heine-Universität Düsseldorf and Friedrich-Schiller-Universität Jena since 1996, this textbook is written mainly for undergraduate and graduate students in computer science, mathematics, and engineering. Researchers, teachers, and practitioners working in these fields also will find this book a comprehensive, up-to-date, research-focused guide to central topics in cryptocomplexity.

Chapter 1 provides more details about this book, including suggestions for how to use it and a brief outline of its chapters.

Acknowledgments

I am deeply indebted to Gerd Wechsung who encouraged me to write this book. I am also much indebted to Tobias Riege, Holger Spakowski, and Gerd Wechsung for carefully proofreading parts of the book and for their invaluable help, corrections, and suggestions. The remaining errors are my responsibility because I couldn't persuade any of the proofreaders to take responsibility for them.

For their generous advice, help, and support over the past decade and for hosting numerous wonderful, inspiring, and prolific research visits at the University of Rochester and the Rochester Institute of Technology, I am deeply indebted to Lane A. Hemaspaandra and Edith Hemaspaandra. I also thank all my fellow researchers, coauthors, and colleagues: Lane and Edith Hemaspaandra, Alina Beygelzimer, Bernd Borchert, Judy Goldsmith, André Große, Christopher M. Homan, Zhigen Jiang, Mitsunori Ogihara, Kari Pasanen, Rajesh P. N. Rao, Tobias Riege, Amitabh Saxena, Holger Spakowski, Jörg Vogel, Osamu Watanabe, and Gerd Wechsung. Some of the research described in this book has been done jointly with them.

I very much appreciate the help and advice of my friends in Rochester, NY, and Atlanta, GA, who had to read some of the stories in this book. I am grateful to Kathleen and Charles Landers-Appell, Mette Stromnes and Dave Lutz, Narin Hassan and Mark Leibert, and, last but not least, Jodi Beckwith and Stefan Cohen.

I gratefully acknowledge inspiring discussions with Klaus Ambos-Spies, Sigurd Assing, Harald Hempel, Uwe Schöning, Andreas Stelzer, Dietrich Stoyan, Klaus W. Wagner, and Gerd Wechsung. For providing a pleasant work environment, I thank my colleagues at the computer science departments in Düsseldorf and Jena: Volker Aurich, Stefan Conrad, Gabor Erdélyi, André Große, Arndt von Haeseler, Harald Hempel, Maren Hinrichs, Dieter Kratsch, Gerhard Lischke, Martin Mauve, Haiko Müller, Tobias Riege, Holger Spakowski, Jörg Vogel, Egon Wanke, and Gerd Wechsung. The support of the technical and secretarial staff in Düsseldorf is also gratefully acknowledged. Thanks to Claudia Forstinger, Claudia Kiometzis, Guido Königstein, Berthold Nöckel, Marga Potthoff, Janus Tomaschewski, and Lutz Voigt.

For generous advice and professional support, I am grateful to the Springer-Verlag series editors and staff, in particular to Wilfried Brauer, Grzegorz Rozenberg, Arto Salomaa, Alfred Hofmann, Ingeborg Mayer, and Ronan Nugent, and to Julia Merz for the cover design. The support of the DFG under grant RO 1202/9-1 and of the DAAD and the NSF under grant NSF-INT-9815095/DAAD-315-PPP-gü-ab, which funded my research projects during the planning and writing of this book, is also gratefully acknowledged.

Above all, I thank my wife, Irene, and my daughters, Paula and Ella, for their love, advice, encouragement, and support. In particular, I thank Irene for her little dragon, ring, and grail, and I thank Paula and Ella for starring in some of the stories told in this book.

Düsseldorf, December 2004 *Jörg Rothe*

Contents

1

Introduction to Cryptocomplexity

About this Book

This book is an introduction to two areas, *complexity theory* and *cryptology*, which are closely related but have developed rather independently of each other. Modern cryptology employs mathematically rigorous concepts and methods of complexity theory. Conversely, current research in complexity theory often is motivated by questions and problems arising in cryptology. This book takes account of this trend, and therefore its subject is what may be dubbed "*cryptocomplexity*," some kind of symbiosis of these two areas.

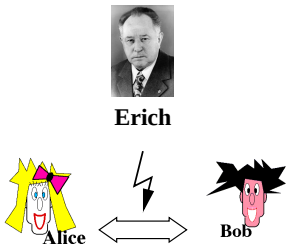

Fig. 1.1. A typical cryptographic scenario

Figure 1.1 shows a typical scenario in cryptography. Alice and Bob wish to exchange messages over an insecure channel such as a public telephone line on which Erich is an eavesdropper. That is why Alice encrypts her messages to Bob in such a way that Bob can easily decrypt them, but Erich cannot. Cryptography is the art and science of designing secure cryptosystems. Alice and Bob use cryptosystems and cryptographic techniques to protect their private data and keep it secret, to electron-

ically sign their messages so that their signatures cannot be forged, for authentication, for the protection of copyrights, to make secure use of computer networks, to exchange information and do business over the internet in a secure way.

Their adversary is Erich, angry that he can intercept or eavesdrop their messages alright, but to no avail for himself. He aims at unauthorized decryption of their ciphertexts, he wants to get his hands at their decryption keys to break their cryptosystem. Cryptanalysis is the art and science of breaking cryptosystems. Cryptology comprises both these fields, cryptography and cryptanalysis.

Cryptography and cryptanalysis fight an ongoing war against each other since ancient times. When our ancestors learned to think and speak and write, they not only seeked to convey their thoughts and messages but also to protect them from unauthorized recipients, i.e., to keep them secret. Already Gaius Julius Caesar, the Roman emperor, was making use of a simple (and easy-to-break) cryptosystem.

Battle after battle has been fought between these two opposing worlds ever since: As soon as the cryptographers have designed a new cryptosystem, the cryptanalysts do not rest before they have broken it, whereupon better cryptosystems are developed, and so forth. The phrases "war" and "battle" can be taken literally here. During World War II, the struggle of the allied codebreakers against the infamous encryption machine *Enigma* used by the Deutsche Wehrmacht was a matter of life and death. The *Enigma*, considered unbreakable at first, was eventually broken by the British codebreakers from Bletchley Park, aided by previous achievements of Polish mathematicians and the cooperation of a German spy. Their achievement was decisive—if not for the war so for a number of battles, especially for the big sea battles and the destruction of the German submarine fleet. Singh [Sin99] and Bauer [Bau00a, Bau00b] elaborately tell the thrilling story of this struggle between the German cryptographers and the allied cryptanalysts. The success of breaking the *Enigma* is attributed to Alan Turing for the most part. His brilliance as a cryptanalyst may be surpassed only by his ingenious, fundamental achievements in theoretical computer science. By inventing the Turing machine, which is named after him, Turing laid the foundations of recursive function and computability theory, the mother of complexity theory.

That efficient algorithms have useful applications in practice is obvious. In contrast, complexity theory aims at proving that certain problems are not efficiently solvable. It provides the means and methods for classifying problems with respect to their inherent computational complexity. It also provides useful tools and techniques for comparing the relative complexity of two given problems via reductions.

In cryptographic settings, provable inefficiency means security: The security of current cryptosystems is based on the assumption that certain problems cannot be solved efficiently. The problem of breaking a cryptosystem can be linked via reductions to suitable problems widely believed to be intractable. Cryptography thus requires and utilizes the computational intractability of problems. In short, cryptography needs and motivates complexity-theoretic notions, models, methods, and results. In particular, the notions of one-way functions, interactive proof systems, and zero-knowledge protocols are central both in cryptology and in complexity theory, which demonstrates the mutual pervasion of these two fields. This book introduces to both cryptology and complexity theory, with a particular focus on their interrelation.

How to Use this Book

This book is based on the author's lectures held at Heinrich-Heine-Universität Düsseldorf and Friedrich-Schiller-Universität Jena since 1996. Written mainly for undergraduate and graduate students in computer science, mathematics, and engineering, it is a valuable source also for researchers, university teachers, and practitioners working in these fields.

This textbook can be used for teaching in more ways than one. On the one hand, it can be used for introductory courses in cryptology from a complexity-theoretic perspective. On the other hand, it can be used for introductory courses in complexity theory, emphasizing potential applications in cryptology. In both regards, this book provides a comprehensive, up-to-date, research-focused guide to the state of the art in these two fields, stressing their connections and choosing a unified approach.

Ideally, however, this book should be used for a series of interrelated courses introducing to both these areas jointly. For example, based on the material presented in this book, a series of four one-semester courses for undergraduate students was test-driven by the author in Düsseldorf. The students' positive feedback suggests that the approach of focusing on the interrelations between complexity theory and cryptology is more profitable for them than teaching these fields separately and independently. A typical course series consists of the following four modules on cryptocomplexity:

Cryptocomplexity I: gives an introduction to complexity theory based on material selected from Chapter 2, Chapter 3, Chapter 5 (e.g., Sections 5.1, 5.2, and 5.6), and Chapter 6 (e.g., Sections 6.1, 6.2, and 6.3).

Cryptocomplexity II: presents more advanced topics from complexity theory based on material selected from Chapter 5 (e.g., Sections 5.3, 5.4, 5.5, and 5.7) and Chapter 6 (e.g., Sections 6.4 and 6.5).

Cryptocomplexity III: gives an introduction to cryptology based on material selected from Chapters 2, 4, and 7.

Cryptocomplexity IV: presents more advanced topics from cryptology based on material selected from Chapters 7 and 8.

Of course, the topics presented in this book can be supplemented by current original research results and other material of interest. Detailed descriptions of these modules can be found at `http://www.cs.uni-duesseldorf.de/~rothe`.

Much care has been taken to motivate and explain the notions and results presented. Numerous examples, figures, and tables are provided to make the text comprehensible, easy-to-read, and hopefully even entertaining at times. Occasionally, before presenting some notion or result in abstract, formal, mathematical terms, it is first introduced and explained by a short story.

Reading this book is not only fun, though, it is also hard work: Every chapter has a set of exercises and problems, with hints at possible solutions or pointers to the original literature. The degree of difficulty of the exercises varies in a broad range; there are rather easy exercises and there are hard ones. Many of the problems are most challenging. Some of them are research problems that were solved only recently in

the literature, and they sometimes require deep insights or clever ideas. Even if they turn out to be too difficult, it is worth trying to solve them.

Due to its comprehensive bibliography (with 508 entries) and subject index (with 1490 main entries), this textbook is also a valuable source for researchers working in complexity theory and cryptology. Starting from scratch and seeking a unified approach, it works its way to the frontiers of current research in selected topics from these two fields. Every chapter concludes with a summary that describes the historical development of the notions and results presented, explains related notions and ideas, and provides comprehensive, detailed bibliographic remarks.

The subject index has an abundance of entries and cross-references because a textbook is only as useful as its index is.[1] Every catchword can have several entries, a boldfaced main entry pointing to its definition, and a number of other entries pointing to theorems containing the catchword. A textbook without an index, or with a poorly or sloppily made index, is of no more help to the reader than a library lacking a classified catalog and having all its books huddled together unsorted. You may stand in front of this huge heap of books, knowing that they contain all the knowledge and the wisdom of the universe, and still you won't be able to find that particular piece of information you are looking for so desperately. This point has been eloquently made by Borges [Bor89] in his short story, "The Library of Babel." By the way, each of the catchwords mentioned in Footnote 1 can indeed be found in the index. Check it out.

Admittedly, this book has a clear focus on theory. Practical aspects of security engineering, such as the creation of secure public-key infrastructures, cannot be found here. A recommendable reference for this topic is Buchmann [Buc01].

In 2003 and 2004, a group of University of Düsseldorf students developed a system that implements a number of cryptosystems, which also are treated in this book. Acknowledgments are due to Tobias Riege, who supervised the students, and to Yves Jerschow, Claudia Lindner, Tim Schlüter, David Schneider, Andreas Stelzer, Philipp Stöcker, Alexander Tchernin, Pavel Tenenbaum, Oleg Umanski, Oliver Wollermann, and Isabel Wolters. The source code in Java can be downloaded from `http://www.cs.uni-duesseldorf.de/~riege/praktikum`.

Overview of the Book Chapters

Chapter 2 provides some background from those fields of computer science and mathematics that are relevant to the topics from complexity theory and cryptology covered in this book. The concepts used are explained with mathematical rigor and in as short a way as possible but to the extent necessary to understand them. In particular, it provides some of the elementary foundations of algorithmics, the theory of formal languages, recursive function theory, logic, algebra, number theory, graph

[1] Suppose you are looking for each occurrence of the phrase *baby cloning* in this book. Or you are interested in a particular tool, say a *chain-saw* or a *Turing machine*. Or you may want to know what this book has to say about *polygamy*, the wizard *Merlin*, the *Ruling Ring*, the *Holy Grail*, or *DNA tests*. Or you may want to learn everything about its *dogmas*.

theory, and probability theory. Although each field is explained from scratch and not much mathematical background is assumed from the reader, some familiarity with the foundations of mathematics and theoretical computer science might be helpful.

In Chapters 3 and 4, the foundations of complexity theory and cryptology are laid, and their historical development is briefly sketched. In Chapter 3, complexity measures and classes are defined in the traditional worst-case model. (The average-case complexity model is not treated here; a useful reference is Wang's excellent survey [Wan97].) Fundamental properties of worst-case complexity are studied, including linear tape-compression and speed-up and the hierarchy theorems for time and space. The relations between the most central complexity classes between logarithmic and polynomial space are explored. Most notable among them are the classes P and NP, deterministic and nondeterministic polynomial time.

P is thought of as a complexity class capturing the intuitive notion of efficient computation, whereas the hardest problems in NP, the NP-complete problems, are thought of as a collection of intractable problems, assuming $P \neq NP$. The P versus NP question, which asks whether or not these two classes differ, is one of the most important open questions in theoretical computer science, and it keeps annoying complexity theorists for more than thirty years now. If $P \neq NP$ then no NP-complete problem can have efficient (i.e., polynomial-time computable) algorithms. On the other hand, if $P = NP$ then all problems in NP are polynomial-time solvable and, in particular, most of the cryptosystems currently in use are broken.

Particular attention is paid in Chapter 3 to complexity-bounded reducibilities, such as the polynomial-time many-one reducibility, and to the related notions of hardness and completeness. Reducibilities are powerful tools for comparing the complexity of two given problems, and completeness captures the hardest problems in a complexity class with respect to a given reducibility. In particular, the complete problems in the classes NL (nondeterministic logarithmic space) and NP are intensely investigated, and a host of specific examples of natural complete problems in these classes are given. These include various variants of the satisfiability problem, which asks whether or not a given boolean formula is satisfiable. The list of problems shown to be NP-complete in this chapter includes certain graph problems, such as the graph three-colorability problem, and certain variants of the knapsack problem. Chapter 8 presents a cryptosystem based on such a knapsack-type problem.

There are problems in NP that seem to be neither NP-complete nor to have efficient algorithms. One such example is the graph isomorphism problem, introduced in Chapter 2 and more deeply studied in Chapters 3, 6, and 8. Another example of a problem that can be solved in nondeterministic polynomial time but is not known to be solvable in deterministic polynomial time is the factoring problem, which will be carefully investigated in Chapter 7. Many cryptosystems, including the famous RSA public-key cryptosystem, are based on the hardness of the factoring problem.

Chapter 3 also introduces an interesting complexity class that seems to lack complete problems: UP, "unambiguous polynomial time," contains exactly those NP problems that never have more than one solution. The complexity class UP is useful for characterizing the existence of certain types of one-way functions in the worst-case model. A function is one-way if it is easy to compute but hard to invert. In

complexity theory, such functions are closely related to Berman and Hartmanis' isomorphism conjecture. One-way functions (in an adequate model of complexity) are also important in cryptography; such functions are discussed in Chapter 8.

Chapter 4 introduces the basic notions of cryptology, such as symmetric (a.k.a. private-key) and asymmtric (a.k.a. public-key) cryptosystems. This chapter presents some classical symmetric cryptosystems, including the substitution, affine, and permutation ciphers, affine linear block ciphers, stream ciphers, the Vigenère, and the Hill cipher. Cryptanalytic attacks on these cryptosystems are provided by example. Moreover, based on the notion of entropy from Shannon's information and coding theory, the notion of perfect secrecy for cryptosystems is introduced and Shannon's result is presented, which provides necessary and sufficient conditions for a cryptosystems to achieve perfect secrecy.

Chapter 5 turns to complexity theory again and introduces hierarchies based on NP, including the boolean hierarchy over NP and the polynomial hierarchy. Relatedly, various polynomial-time Turing reducibilities are defined. Both these hierarchies contain NP as their first level and are very useful to classify important problems that seem to be harder than NP-complete problems. Examples of problems complete in the higher levels of the boolean hierarchy are the "exact" variants of NP-complete optimization problems, facet problems, and critical graph problems. Examples of problems complete in the higher levels of the polynomial hierarchy are certain variants of NP-complete problems that can be represented by a bounded number of alternating polynomially length-bounded quantifiers. The canonical example of such problems is the quantified boolean formula problem with a bounded number of alternating quantifiers, which generalizes the satisfiability problem.

Relatedly, the notion of alternating Turing machines is introduced in Chapter 5, and P and PSPACE are characterized in terms of such machines: Deterministic polynomial time equals alternating logarithmic space, and deterministic polynomial space equals alternating polynomial time. The former result shows that alternating Turing machines are a reasonable model of parallel computation, since they satisfy Cook's criterion that parallel time is roughly the same as sequential (i.e., deterministic) space. The latter result shows that the quantified boolean formula problem with an unbounded number of alternating quantifiers is complete for PSPACE.

There is a remarkable connection between the polynomial hierarchy and the boolean hierarchy over NP: If the boolean hierarchy collapses to a finite level, then so does the polynomial hierarchy. Chapter 5 further introduces the query hierarchies over NP with a bounded number of queries, and the low and high hierarchies within NP. The low hierarchy can be used to measure the complexity of NP problems that seem to be neither in P nor NP-complete.

Chapter 6 is concerned with randomized algorithms and probabilistic complexity classes. In particular, a randomized algorithm for the NP-complete satisfiability problem is introduced, which still runs in exponential time but is faster than the naive deterministic algorithm for this problem. Moreover, Monte Carlo and Las Vegas algorithms and the probabilistic complexity classes PP (probabilistic polynomial time), RP (random polynomial time), ZPP (zero-error probabilistic polynomial time), and BPP (bounded-error probabilistic polynomial time) are introduced and thoroughly

studied in Chapter 6. Bounding the error away from one half yields a very useful probability amplification by which the error in the computation can be made exponentially small in the input size. Such a small error probability can be safely neglected for most practical applications. Again, some of the probabilistic complexity classes (e.g., PP) do have complete problems, whereas others (e.g., BPP) are unlikely to have complete problems.

Chapter 6 also studies the Arthur-Merlin games introduced by Babai and Moran. Arthur-Merlin games can be regarded as interactive proof systems with public coin tosses, and they can be used to define a hierarchy of complexity classes. The main results about the Arthur-Merlin hierarchy in Chapter 6 are, first, that this hierarchy collapses to a finite level, and, second, that the graph isomorphism problem is contained in the second level of this hierarchy. Consequently, the graph isomorphism problem is contained in the low hierarchy and thus is unlikely to be NP-complete.

Chapter 7 introduces the RSA cryptosystem, the first public-key cryptosystem developed in the public sector, which still is widely used in practice today. The RSA digital signature scheme, which is based on the RSA public-key cryptosystem, is also presented. A digital signature protocol enables Alice to sign her messages to Bob so that Bob can verify that indeed she was the sender, and without Erich being able to forge Alice's signature. In addition, numerous cryptanalytic attacks on the RSA system are surveyed and thoroughly discussed, and for each attack on RSA presented, possible countermeasures are suggested.

Related to the RSA system, Chapter 7 investigates the factoring problem and the primality problem in depth. On the one hand, the security of RSA crucially depends on the presumed hardness of factoring large integers. On the other hand, the RSA cryptosystem and digital signature scheme both require the efficient generation of large primes, as do many other cryptosystems. The complexity of the most prominent factoring methods known, such as the quadratic sieve, is discussed in Chapter 7. Note that the factoring problem is currently known neither to have an efficient algorithm nor to have a rigorous proof of its hardness.

Chapter 7 further presents a number of efficient primality tests that are used in practice, including the Fermat test, the Miller–Rabin test, and the Solovay–Strassen test. These are randomized algorithms, and some of them are of the Monte Carlo type. A recent result showing that the primality problem can be solved in deterministic polynomial time is also discussed.

Chapter 8 surveys further important public-key cryptosystems and cryptographic protocols, including the Diffie–Hellman secret-key agreement protocol and the ElGamal digital signature protocol. The latter protocol, with appropriate modifications, has been adopted as the United States digital signature standard. Relatedly, the discrete logarithm problem is carefully studied in this chapter. The security of many important protocols, such as the two just mentioned, relies on the presumed hardness of the discrete logarithm problem.

Revisiting the graph isomorphism problem and the notion of Arthur-Merlin games that were studied in previous chapters, Chapter 8 introduces the notion of zero-knowledge protocols, which is related to the cryptographic task of authentication.

There have been attempts in the past to base cryptosystem on NP-hard problems; in particular, on variants of the knapsack problem. Some of those cryptosystems were broken, whereas others are still unbroken. One such cryptosystem is presented and critically discussed in Chapter 8. Relatedly, the notion of a trapdoor one-way function, which is important in public-key cryptography, is discussed. Finally, this chapter introduces protocols for secret-key agreement and digital signatures that are based on associative, strongly noninvertible one-way functions (in the worst-case model).

Obviously, there are many interesting topics and results in complexity theory and cryptology that could not be covered in this book. Here are some recommendable references. For example, approximation and nonapproximation results, which are of both theoretical and practical importance, are not covered here; see, e.g., Ausiello et al.[ACG+03], Vazirani [Vaz03], and the comprehensive, up-to-date compendium of NP optimization problems edited by Crescenzi, Kann, Halldórsson, Karpinski, and Woeginger: `http://www.nada.kth.se/~viggo/problemlist/compendium.html`.

For a variety of further important topics of complexity theory, see the books by Balcázar, Díaz, and Gabarró [BDG95, BDG90], Bovet and Crescenzi [BC93], Du and Ko [DK00], Garey and Johnson [GJ79], L. Hemaspaandra and Ogihara [HO02], Papadimitriou [Pap94], Reischuk [Rei90], Vollmer [Vol99], Wagner and Wechsung [WW86, Wec00], and Wegener [Weg87, Weg03], and the collections edited by Selman and L. Hemaspaandra [Sel90, HS97] and Ambos-Spies, Homer, and Schöning [AHS93]. For topics of cryptology not covered here, see, e.g., Goldreich [Gol99, Gol01], Luby [Lub96], Micciancio and Goldwasser [MG02], Salomaa [Sal96], Schneier [Sch96], Stinson [Sti02], and Welsh [Wel98].