

信息安全丛书

密码学基础

英文版

Foundations of Cryptography
Basic Tools

[以色列] Oded Goldreich 著



电子工业出版社
Publishing House of Electronics Industry
www.phei.com.cn

信息安全丛书

密码学基础

(英文版)

Foundations of Cryptography
Basic Tools

[以色列] Oded Goldreich 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

密码学涉及解决通信保密问题的计算系统的概念、定义及构造。密码系统的设计必须基于坚实的基础。本书对这一基础问题给出了系统而严格的论述：用已有工具来定义密码系统的目标并解决新的密码学问题。全书集中讨论了基本的数学工具：计算困难性（单向函数）、伪随机性以及零知识证明等。本书的重点是澄清基本概念及证明密码学问题解决方法的可行性，而不侧重于对特殊方法的描述。

本书可作为密码学、应用数学、信息安全等专业的研究生教材，也可作为相关专业人员的参考用书。

Originally published by Cambridge University Press in 2001.

This reprint edition is published with the permission of the Syndicate of the Press of the University of Cambridge, Cambridge, England.

本书原版由 Cambridge University Press 2001 年出版。

本书英文影印版权得到英国剑桥大学出版社授权。

© Oded Goldreich 2001.

THIS EDITION IS LICENSED FOR DISTRIBUTION AND SALE IN THE PEOPLE'S REPUBLIC OF CHINA ONLY EXCLUDING HONG KONG, TAIWAN AND MACAU AND MAY NOT BE DISTRIBUTED AND SOLD ELSEWHERE.

本书英文影印版仅限于在中华人民共和国境内发行与销售（不包括香港、台湾和澳门地区），并不得在其他地区发行与销售。

本书英文影印版权（仅限中国大陆）由 Cambridge University Press 授予电子工业出版社。其原文版权受法律保护。未经许可，不得以任何形式或手段复制或抄袭本书内容。

版权贸易合同登记号：图字：01-2002-2419

图书在版编目（CIP）数据

密码学基础（英文版）/（以色列）戈德赖克（Goldreich, O.）著. -北京：电子工业出版社，2003.1

（信息安全丛书）

书名原文：Foundations of Cryptography: Basic Tools

ISBN 7-5053-8178-4

I. 密… II. 戈… III. 密码-理论-英文 IV. TN918.1

中国版本图书馆CIP数据核字（2002）第086173号

责任编辑：李秦华 陶淑毅

印刷者：北京市增富印刷有限责任公司

出版发行：电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编：100036

经 销：各地新华书店

开 本：787 × 980 1/16 印张：24.5 字数：564千字

版 次：2003年1月第1版 2003年1月第1次印刷

定 价：39.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077

Preface

*It is possible to build a cabin with no foundations,
but not a lasting building.*

Eng. Isidor Goldreich (1906–1995)

Cryptography is concerned with the construction of schemes that should be able to withstand any abuse. Such schemes are constructed so as to maintain a desired functionality, even under malicious attempts aimed at making them deviate from their prescribed functionality.

The design of cryptographic schemes is a very difficult task. One cannot rely on intuitions regarding the typical state of the environment in which a system will operate. For sure, an *adversary* attacking the system will try to manipulate the environment into untypical states. Nor can one be content with countermeasures designed to withstand specific attacks, because the adversary (who will act after the design of the system has been completed) will try to attack the schemes in ways that typically will be different from the ones the designer envisioned. Although the validity of the foregoing assertions seems self-evident, still some people hope that, in practice, ignoring these tautologies will not result in actual damage. Experience shows that such hopes are rarely met; cryptographic schemes based on make-believe are broken, typically sooner rather than later.

In view of the foregoing, we believe that it makes little sense to make assumptions regarding the specific *strategy* that an adversary may use. The only assumptions that can be justified refer to the computational *abilities* of the adversary. Furthermore, it is our opinion that the design of cryptographic systems has to be based on *firm foundations*, whereas ad hoc approaches and heuristics are a very dangerous way to go. A heuristic may make sense when the designer has a very good idea about the environment in which a scheme is to operate, but a cryptographic scheme will have to operate in a maliciously selected environment that typically will transcend the designer's view.

This book is aimed at presenting firm foundations for cryptography. The foundations of cryptography are the paradigms, approaches, and techniques used to conceptualize, define, and provide solutions to natural "security concerns." We shall present some of these paradigms, approaches, and techniques, as well as some of the fundamental results

obtained by using them. Our emphasis is on the clarification of fundamental concepts and on demonstrating the feasibility of solving several central cryptographic problems.

Solving a cryptographic problem (or addressing a security concern) is a two-stage process consisting of a *definitional stage* and a *constructive stage*. First, in the definitional stage, the functionality underlying the natural concern must be identified and an adequate cryptographic problem must be defined. Trying to list all undesired situations is infeasible and prone to error. Instead, one should define the functionality in terms of operation in an imaginary ideal model and then require a candidate solution to emulate this operation in the real, clearly defined model (which will specify the adversary's abilities). Once the definitional stage is completed, one proceeds to construct a system that will satisfy the definition. Such a construction may use some simpler tools, and its security is to be proved relying on the features of these tools. (In practice, of course, such a scheme also may need to satisfy some specific efficiency requirements.)

This book focuses on several archetypical cryptographic problems (e.g., encryption and signature schemes) and on several central tools (e.g., computational difficulty, pseudorandomness, and zero-knowledge proofs). For each of these problems (resp., tools), we start by presenting the natural concern underlying it (resp., its intuitive objective), then define the problem (resp., tool), and finally demonstrate that the problem can be solved (resp., the tool can be constructed). In the last step, our focus is on demonstrating the feasibility of solving the problem, not on providing a practical solution. As a secondary concern, we typically discuss the level of practicality (or impracticality) of the given (or known) solution.

Computational Difficulty

The specific constructs mentioned earlier (as well as most constructs in this area) can exist only if some sort of computational hardness (i.e., difficulty) exists. Specifically, all these problems and tools require (either explicitly or implicitly) the ability to generate instances of hard problems. Such ability is captured in the definition of one-way functions (see further discussion in Section 2.1). Thus, one-way functions are the very minimum needed for doing most sorts of cryptography. As we shall see, they actually suffice for doing much of cryptography (and the rest can be done by augmentations and extensions of the assumption that one-way functions exist).

Our current state of understanding of efficient computation does not allow us to prove that one-way functions exist. In particular, the existence of one-way functions implies that \mathcal{NP} is not contained in $\mathcal{BPP} \supseteq \mathcal{P}$ (not even "on the average"), which would resolve the most famous open problem of computer science. Thus, we have no choice (at this stage of history) but to assume that one-way functions exist. As justification for this assumption we can only offer the combined beliefs of hundreds (or thousands) of researchers. Furthermore, these beliefs concern a simply stated assumption, and their validity is supported by several widely believed conjectures that are central to some fields (e.g., the conjecture that factoring integers is difficult is central to computational number theory).

As we need assumptions anyhow, why not just assume what we want, that is, the existence of a solution to some natural cryptographic problem? Well, first we need

to know what we want: As stated earlier, we must first clarify what exactly we do want; that is, we must go through the typically complex definitional stage. But once this stage is completed, can we just assume that the definition derived can be met? Not really: The mere fact that a definition has been derived does *not* mean that it can be met, and one can easily define objects that cannot exist (without this fact being obvious in the definition). The way to demonstrate that a definition is viable (and so the intuitive security concern can be satisfied at all) is to construct a solution based on a *better-understood* assumption (i.e., one that is more common and widely believed). For example, looking at the definition of zero-knowledge proofs, it is not a priori clear that such proofs exist at all (in a non-trivial sense). The non-triviality of the notion was first demonstrated by presenting a zero-knowledge proof system for statements regarding Quadratic Residuosity that are believed to be difficult to verify (without extra information). Furthermore, contrary to prior belief, it was later shown that the existence of one-way functions implies that any \mathcal{NP} -statement can be proved in zero-knowledge. Thus, facts that were not at all known to hold (and were even believed to be false) were shown to hold by reduction to widely believed assumptions (without which most of modern cryptography would collapse anyhow). To summarize, not all assumptions are equal, and so reducing a complex, new, and doubtful assumption to a widely believed simple (or even merely simpler) assumption is of great value. Furthermore, reducing the solution of a new task to the assumed security of a well-known primitive task typically means providing a construction that, using the known primitive, will solve the new task. This means that we not only know (or assume) that the new task is solvable but also have a solution based on a primitive that, being well known, typically has several candidate implementations.

Structure and Prerequisites

Our aim is to present the basic concepts, techniques, and results in cryptography. As stated earlier, our emphasis is on the clarification of fundamental concepts and the relationships among them. This is done in a way independent of the particularities of some popular number-theoretic examples. These particular examples played a central role in the development of the field and still offer the most practical implementations of all cryptographic primitives, but this does not mean that the presentation has to be linked to them. On the contrary, we believe that concepts are best clarified when presented at an abstract level, decoupled from specific implementations. Thus, the most relevant background for this book is provided by basic knowledge of algorithms (including randomized ones), computability, and elementary probability theory. Background on (computational) number theory, which is required for specific implementations of certain constructs, is not really required here (yet a short appendix presenting the most relevant facts is included in this volume so as to support the few examples of implementations presented here).

Organization of the work. This work is organized into three parts (see Figure 0.1), to be presented in three volumes: *Basic Tools*, *Basic Applications*, and *Beyond the Basics*. This first volume contains an introductory chapter as well as the first part

Volume 1: Introduction and Basic Tools
Chapter 1: Introduction
Chapter 2: Computational Difficulty (One-Way Functions)
Chapter 3: Pseudorandom Generators
Chapter 4: Zero-Knowledge Proof Systems
Volume 2: Basic Applications
Chapter 5: Encryption Schemes
Chapter 6: Signature Schemes
Chapter 7: General Cryptographic Protocols
Volume 3: Beyond the Basics
...

Figure 0.1: Organization of the work.

(basic tools). It provides chapters on computational difficulty (one-way functions), pseudorandomness, and zero-knowledge proofs. These basic tools will be used for the basic applications in the second volume, which will consist of encryption, signatures, and general cryptographic protocols.

The partition of the work into three volumes is a logical one. Furthermore, it offers the advantage of publishing the first part without waiting for the completion of the other parts. Similarly, we hope to complete the second volume within a couple of years and publish it without waiting for the third volume.

Organization of this first volume. This first volume consists of an introductory chapter (Chapter 1), followed by chapters on computational difficulty (one-way functions), pseudorandomness, and zero-knowledge proofs (Chapters 2–4, respectively). Also included are two appendixes, one of them providing a brief summary of Volume 2. Figure 0.2 depicts the high-level structure of this first volume.

Historical notes, suggestions for further reading, some open problems, and some exercises are provided at the end of each chapter. The exercises are *mostly* designed to assist and test one's basic understanding of the main text, not to test or inspire creativity. The open problems are fairly well known; still, we recommend that one check their current status (e.g., at our updated-notices web site).

Web site for notices regarding this book. We intend to maintain a web site listing corrections of various types. The location of the site is

<http://www.wisdom.weizmann.ac.il/~oded/foc-book.html>

Using This Book

The book is intended to serve as both a textbook and a reference text. That is, it is aimed at serving both the beginner and the expert. In order to achieve that goal, the presentation of the basic material is very detailed, so as to allow a typical undergraduate in computer science to follow it. An advanced student (and certainly an expert) will find the pace in these parts far too slow. However, an attempt has been made to allow the latter reader to easily skip details that are obvious to him or her. In particular, proofs typically are presented in a modular way. We start with a high-level sketch of the main ideas and

Chapter 1: <i>Introduction</i>
Main topics covered by the book (Sec. 1.1)
Background on probability and computation (Sec. 1.2 and 1.3)
Motivation to the rigorous treatment (Sec. 1.4)
Chapter 2: <i>Computational Difficulty (One-Way Functions)</i>
Motivation and definitions (Sec. 2.1 and 2.2)
One-way functions: weak implies strong (Sec. 2.3)
Variants (Sec. 2.4) and advanced material (Sec. 2.6)
Hard-core predicates (Sec. 2.5)
Chapter 3: <i>Pseudorandom Generators</i>
Motivation and definitions (Sec. 3.1–3.3)
Constructions based on one-way permutations (Sec. 3.4)
Pseudorandom functions (Sec. 3.6)
Advanced material (Sec. 3.5 and 3.7)
Chapter 4: <i>Zero-Knowledge Proofs</i>
Motivation and definitions (Sec. 4.1–4.3)
Zero-knowledge proofs for \mathcal{NP} (Sec. 4.4)
Advanced material (Sec. 4.5–4.11)
Appendix A: Background in Computational Number Theory
Appendix B: Brief Outline of Volume 2
Bibliography and Index

Figure 0.2: Rough organization of this volume.

only later pass to the technical details. The transition from high-level description to lower-level details is typically indicated by phrases such as “details follow.”

In a few places, we provide straightforward but tedious details in indented paragraphs such as this one. In some other (even fewer) places, such paragraphs provide technical proofs of claims that are of marginal relevance to the topic of the book.

More advanced material typically is presented at a faster pace and with fewer details. Thus, we hope that the attempt to satisfy a wide range of readers will not harm any of them.

Teaching. The material presented in this book is, on one hand, way beyond what one may want to cover in a semester course, and on the other hand it falls very short of what one may want to know about cryptography in general. To assist these conflicting needs, we make a distinction between *basic* and *advanced* material and provide suggestions for further reading (in the last section of each chapter). In particular, those sections marked by an asterisk are intended for advanced reading.

Volumes 1 and 2 of this work are intended to provide all the material needed for a course on the foundations of cryptography. For a one-semester course, the instructor definitely will need to skip all advanced material (marked by asterisks) and perhaps even some basic material; see the suggestions in Figure 0.3. This should allow, depending on the class, coverage of the basic material at a reasonable level (i.e., all material marked as “main” and some of the “optional”). Volumes 1 and 2 can also serve as a textbook for a two-semester course. Either way, this first volume covers only the first half of the material for such a course. The second half will be covered in Volume 2. Meanwhile,

Each lecture consists of one hour. Lectures 1–15 are covered by this first volume. Lectures 16–28 will be covered by the second volume.

Lecture 1: Introduction, background, etc.
(depending on class)

Lectures 2–5: *Computational Difficulty (One-Way Functions)*

Main: Definition (Sec. 2.2), Hard-core predicates (Sec. 2.5)

Optional: Weak implies strong (Sec. 2.3), and Sec. 2.4.2–2.4.4

Lectures 6–10: *Pseudorandom Generators*

Main: Definitional issues and a construction (Sec. 3.2–3.4)

Optional: Pseudorandom functions (Sec. 3.6)

Lectures 11–15: *Zero-Knowledge Proofs*

Main: Some definitions and a construction (Sec. 4.2.1, 4.3.1, 4.4.1–4.4.3)

Optional: Sec. 4.2.2, 4.3.2, 4.3.3, 4.3.4, 4.4.4

Lectures 16–20: *Encryption Schemes*

Definitions and a construction (consult Appendix B.1.1–B.1.2)

(See also fragments of a draft for the encryption chapter [99].)

Lectures 21–24: *Signature Schemes*

Definition and a construction (consult Appendix B.2)

(See also fragments of a draft for the signatures chapter [100].)

Lectures 25–28: *General Cryptographic Protocols*

The definitional approach and a general construction (sketches).

(Consult Appendix B.3; see also [98].)

Figure 0.3: Plan for one-semester course on the foundations of cryptography.

we suggest the use of other sources for the second half. A brief summary of Volume 2 and recommendations for alternative sources are given in Appendix B. (In addition, fragments and/or preliminary drafts for the three chapters of Volume 2 are available from earlier texts, [99], [100], and [98], respectively.)

A course based solely on the material in this first volume is indeed possible, but such a course cannot be considered a stand-alone course in cryptography because this volume does not consider at all the basic tasks of encryption and signatures.

Practice. The aim of this work is to provide sound theoretical foundations for cryptography. As argued earlier, such foundations are necessary for any *sound* practice of cryptography. Indeed, sound practice requires more than theoretical foundations, whereas this work makes no attempt to provide anything beyond the latter. However, given sound foundations, one can learn and evaluate various practical suggestions that appear elsewhere (e.g., [158]). On the other hand, the absence of sound foundations will result in inability to critically evaluate practical suggestions, which in turn will lead to unsound decisions. Nothing could be more harmful to the design of schemes that need to withstand adversarial attacks than misconceptions about such attacks.

Relationship to another book by the author. A frequently asked question concerns the relationship of this work to my text *Modern Cryptography, Probabilistic Proofs and Pseudorandomness* [97]. That text consists of three brief introductions to the related topics in the title. Specifically, in Chapter 1 it provides a brief (i.e., 30-page)

summary of this work. The other two chapters of *Modern Cryptography, Probabilistic Proofs and Pseudorandomness* [97] provide a wider perspective on two topics mentioned in this volume (i.e., probabilistic proofs and pseudorandomness). Further comments on the latter aspect are provided in the relevant chapters of this volume.

Acknowledgments

First of all, I would like to thank three remarkable people who had a tremendous influence on my professional development: Shimon Even introduced me to theoretical computer science and closely guided my first steps. Silvio Micali and Shafi Goldwasser led my way in the evolving foundations of cryptography and shared with me their ongoing efforts toward further development of those foundations.

I have collaborated with many researchers, but I feel that my work with Benny Chor and Avi Wigderson has had the most important impact on my professional development and career. I would like to thank them both for their indispensable contributions to our joint research and for the excitement and pleasure of working with them.

Leonid Levin deserves special thanks as well. I have had many interesting discussions with Leonid over the years, and sometimes it has taken me too long to realize how helpful those discussions have been.

Next, I would like to thank a few colleagues and friends with whom I have had significant interactions regarding cryptography and related topics. These include Noga Alon, Boaz Barak, Mihir Bellare, Ran Canetti, Ivan Damgard, Uri Feige, Shai Halevi, Johan Hastad, Amir Herzberg, Russell Impagliazzo, Joe Kilian, Hugo Krawczyk, Eyal Kushilevitz, Yehuda Lindell, Mike Luby, Daniele Micciancio, Moni Naor, Noam Nisan, Andrew Odlyzko, Yair Oren, Rafail Ostrovsky, Erez Petrank, Birgit Pfitzmann, Omer Reingold, Ron Rivest, Amit Sahai, Claus Schnorr, Adi Shamir, Victor Shoup, Madhu Sudan, Luca Trevisan, Salil Vadhan, Ronen Vainish, Yacob Yacobi, and David Zuckerman.

Even assuming I have not overlooked people with whom I have had significant interactions on topics related to this book, the complete list of people to whom I am indebted is far more extensive. It certainly includes the authors of many papers mentioned in the Bibliography. It also includes the authors of many cryptography-related papers that I have not cited and the authors of many papers regarding the theory of computation at large (a theory taken for granted in this book).

Finally, I would like to thank Alon Rosen for carefully reading this manuscript and suggesting numerous corrections.

目录概览

第 1 章 引言	1
Introduction	
第 2 章 计算困难性	30
Computational Difficulty	
第 3 章 伪随机生成器	101
Pseudorandom Generators	
第 4 章 零知识证明系统	184
Zero-Knowledge Proof Systems	
附录 A 计算数论基础	331
Background in Computational Number Theory	
附录 B 第二卷概述	338
Brief Outline of Volume 2	
参考文献	355
Bibliography	
索引	367
Index	

Contents

1	Introduction	1
1.1.	Cryptography: Main Topics	1
1.1.1.	Encryption Schemes	2
1.1.2.	Pseudorandom Generators	3
1.1.3.	Digital Signatures	4
1.1.4.	Fault-Tolerant Protocols and Zero-Knowledge Proofs	6
1.2.	Some Background from Probability Theory	8
1.2.1.	Notational Conventions	8
1.2.2.	Three Inequalities	9
1.3.	The Computational Model	12
1.3.1.	\mathcal{P} , \mathcal{NP} , and \mathcal{NP} -Completeness	12
1.3.2.	Probabilistic Polynomial Time	13
1.3.3.	Non-Uniform Polynomial Time	16
1.3.4.	Intractability Assumptions	19
1.3.5.	Oracle Machines	20
1.4.	Motivation to the Rigorous Treatment	21
1.4.1.	The Need for a Rigorous Treatment	21
1.4.2.	Practical Consequences of the Rigorous Treatment	23
1.4.3.	The Tendency to Be Conservative	24
1.5.	Miscellaneous	25
1.5.1.	Historical Notes	25
1.5.2.	Suggestions for Further Reading	27
1.5.3.	Open Problems	27
1.5.4.	Exercises	28

2	Computational Difficulty	30
2.1.	One-Way Functions: Motivation	31
2.2.	One-Way Functions: Definitions	32
2.2.1.	Strong One-Way Functions	32
2.2.2.	Weak One-Way Functions	35
2.2.3.	Two Useful Length Conventions	35
2.2.4.	Candidates for One-Way Functions	40
2.2.5.	Non-Uniformly One-Way Functions	41
2.3	Weak One-Way Functions Imply Strong Ones	43
2.3.1.	The Construction and Its Analysis (Proof of Theorem 2.3.2)	44
2.3.2.	Illustration by a Toy Example	48
2.3.3.	Discussion	50
2.4.	One-Way Functions: Variations	51
2.4.1.*	Universal One-Way Function	52
2.4.2.	One-Way Functions as Collections	53
2.4.3.	Examples of One-Way Collections	55
2.4.4.	Trapdoor One-Way Permutations	58
2.4.5.*	Claw-Free Functions	60
2.4.6.*	On Proposing Candidates	63
2.5.	Hard-Core Predicates	64
2.5.1.	Definition	64
2.5.2.	Hard-Core Predicates for Any One-Way Function	65
2.5.3.*	Hard-Core Functions	74
2.6.*	Efficient Amplification of One-Way Functions	78
2.6.1.	The Construction	80
2.6.2.	Analysis	81
2.7.	Miscellaneous	88
2.7.1.	Historical Notes	89
2.7.2.	Suggestions for Further Reading	89
2.7.3.	Open Problems	91
2.7.4.	Exercises	92
3	Pseudorandom Generators	101
3.1.	Motivating Discussion	102
3.1.1.	Computational Approaches to Randomness	102
3.1.2.	A Rigorous Approach to Pseudorandom Generators	103
3.2.	Computational Indistinguishability	103
3.2.1.	Definition	104
3.2.2.	Relation to Statistical Closeness	106
3.2.3.	Indistinguishability by Repeated Experiments	107
3.2.4.*	Indistinguishability by Circuits	111
3.2.5.	Pseudorandom Ensembles	112
3.3.	Definitions of Pseudorandom Generators	112
3.3.1.	Standard Definition of Pseudorandom Generators	113

3.3.2.	Increasing the Expansion Factor	114
3.3.3.*	Variable-Output Pseudorandom Generators	118
3.3.4.	The Applicability of Pseudorandom Generators	119
3.3.5.	Pseudorandomness and Unpredictability	119
3.3.6.	Pseudorandom Generators Imply One-Way Functions	123
3.4.	Constructions Based on One-Way Permutations	124
3.4.1.	Construction Based on a Single Permutation	124
3.4.2.	Construction Based on Collections of Permutations	131
3.4.3.*	Using Hard-Core Functions Rather than Predicates	134
3.5.*	Constructions Based on One-Way Functions	135
3.5.1.	Using 1-1 One-Way Functions	135
3.5.2.	Using Regular One-Way Functions	141
3.5.3.	Going Beyond Regular One-Way Functions	147
3.6.	Pseudorandom Functions	148
3.6.1.	Definitions	148
3.6.2.	Construction	150
3.6.3.	Applications: A General Methodology	157
3.6.4.*	Generalizations	158
3.7.*	Pseudorandom Permutations	164
3.7.1.	Definitions	164
3.7.2.	Construction	166
3.8.	Miscellaneous	169
3.8.1.	Historical Notes	169
3.8.2.	Suggestions for Further Reading	170
3.8.3.	Open Problems	172
3.8.4.	Exercises	172
4	Zero-Knowledge Proof Systems	184
4.1.	Zero-Knowledge Proofs: Motivation	185
4.1.1.	The Notion of a Proof	187
4.1.2.	Gaining Knowledge	189
4.2.	Interactive Proof Systems	190
4.2.1.	Definition	190
4.2.2.	An Example (Graph Non-Isomorphism in IP)	195
4.2.3.*	The Structure of the Class IP	198
4.2.4.	Augmentation of the Model	199
4.3.	Zero-Knowledge Proofs: Definitions	200
4.3.1.	Perfect and Computational Zero-Knowledge	200
4.3.2.	An Example (Graph Isomorphism in PZK)	207
4.3.3.	Zero-Knowledge with Respect to Auxiliary Inputs	213
4.3.4.	Sequential Composition of Zero-Knowledge Proofs	216
4.4.	Zero-Knowledge Proofs for \mathcal{NP}	223
4.4.1.	Commitment Schemes	223
4.4.2.	Zero-Knowledge Proof of Graph Coloring	228

4.4.3.	The General Result and Some Applications	240
4.4.4.	Second-Level Considerations	243
4.5.*	Negative Results	246
4.5.1.	On the Importance of Interaction and Randomness	247
4.5.2.	Limitations of Unconditional Results	248
4.5.3.	Limitations of Statistical ZK Proofs	250
4.5.4.	Zero-Knowledge and Parallel Composition	251
4.6.*	Witness Indistinguishability and Hiding	254
4.6.1.	Definitions	254
4.6.2.	Parallel Composition	258
4.6.3.	Constructions	259
4.6.4.	Applications	261
4.7.*	Proofs of Knowledge	262
4.7.1.	Definition	262
4.7.2.	Reducing the Knowledge Error	267
4.7.3.	Zero-Knowledge Proofs of Knowledge for \mathcal{NP}	268
4.7.4.	Applications	269
4.7.5.	Proofs of Identity (Identification Schemes)	270
4.7.6.	Strong Proofs of Knowledge	274
4.8.*	Computationally Sound Proofs (Arguments)	277
4.8.1.	Definition	277
4.8.2.	Perfectly Hiding Commitment Schemes	278
4.8.3.	Perfect Zero-Knowledge Arguments for \mathcal{NP}	284
4.8.4.	Arguments of Poly-Logarithmic Efficiency	286
4.9.*	Constant-Round Zero-Knowledge Proofs	288
4.9.1.	Using Commitment Schemes with Perfect Secrecy	289
4.9.2.	Bounding the Power of Cheating Provers	294
4.10.*	Non-Interactive Zero-Knowledge Proofs	298
4.10.1.	Basic Definitions	299
4.10.2.	Constructions	300
4.10.3.	Extensions	306
4.11.*	Multi-Prover Zero-Knowledge Proofs	311
4.11.1.	Definitions	311
4.11.2.	Two-Sender Commitment Schemes	313
4.11.3.	Perfect Zero-Knowledge for \mathcal{NP}	317
4.11.4.	Applications	319
4.12.	Miscellaneous	320
4.12.1.	Historical Notes	320
4.12.2.	Suggestions for Further Reading	322
4.12.3.	Open Problems	323
4.12.4.	Exercises	323
Appendix A: Background in Computational Number Theory		331
A.1.	Prime Numbers	331
A.1.1.	Quadratic Residues Modulo a Prime	331

A.1.2.	Extracting Square Roots Modulo a Prime	332
A.1.3.	Primality Testers	332
A.1.4.	On Uniform Selection of Primes	333
A.2.	Composite Numbers	334
A.2.1.	Quadratic Residues Modulo a Composite	335
A.2.2.	Extracting Square Roots Modulo a Composite	335
A.2.3.	The Legendre and Jacobi Symbols	336
A.2.4.	Blum Integers and Their Quadratic-Residue Structure	337
Appendix B: Brief Outline of Volume 2		338
B.1.	Encryption: Brief Summary	338
B.1.1.	Definitions	338
B.1.2.	Constructions	340
B.1.3.	Beyond Eavesdropping Security	343
B.1.4.	Some Suggestions	345
B.2.	Signatures: Brief Summary	345
B.2.1.	Definitions	346
B.2.2.	Constructions	347
B.2.3.	Some Suggestions	349
B.3.	Cryptographic Protocols: Brief Summary	350
B.3.1.	Definitions	350
B.3.2.	Constructions	352
B.3.3.	Some Suggestions	353
	Bibliography	355
	Index	367

Note: Asterisks throughout Contents indicate advanced material.

Introduction

In this chapter we briefly discuss the goals of cryptography (Section 1.1). In particular, we discuss the basic problems of secure encryption, digital signatures, and fault-tolerant protocols. These problems lead to the notions of pseudorandom generators and zero-knowledge proofs, which are discussed as well.

Our approach to cryptography is based on computational complexity. Hence, this introductory chapter also contains a section presenting the computational models used throughout the book (Section 1.3). Likewise, this chapter contains a section presenting some elementary background from probability theory that is used extensively in the book (Section 1.2).

Finally, we motivate the rigorous approach employed throughout this book and discuss some of its aspects (Section 1.4).

Teaching Tip. Parts of Section 1.4 may be more suitable for the last lecture (i.e., as part of the concluding remarks) than for the first one (i.e., as part of the introductory remarks). This refers specifically to Sections 1.4.2 and 1.4.3.

1.1. Cryptography: Main Topics

Historically, the term “cryptography” has been associated with the problem of designing and analyzing *encryption schemes* (i.e., schemes that provide secret communication over insecure communication media). However, since the 1970s, problems such as constructing unforgeable *digital signatures* and designing *fault-tolerant protocols* have also been considered as falling within the domain of cryptography. In fact, cryptography can be viewed as concerned with the design of any system that needs to withstand malicious attempts to abuse it. Furthermore, cryptography as redefined here makes essential use of some tools that need to be treated in a book on the subject. Notable examples include one-way functions, pseudorandom generators, and zero-knowledge proofs. In this section we briefly discuss these terms.