
Contents

Preface	xv
I Introduction and Classical Cryptography	
1 Introduction	3
1.1 Cryptography and Modern Cryptography	3
1.2 The Setting of Private-Key Encryption	4
1.3 Historical Ciphers and Their Cryptanalysis	8
1.4 Principles of Modern Cryptography	16
1.4.1 Principle 1 – Formal Definitions	17
1.4.2 Principle 2 – Precise Assumptions	20
1.4.3 Principle 3 – Proofs of Security	22
1.4.4 Provable Security and Real-World Security	22
References and Additional Reading	23
Exercises	24
2 Perfectly Secret Encryption	25
2.1 Definitions	26
2.2 The One-Time Pad	32
2.3 Limitations of Perfect Secrecy	35
2.4 *Shannon’s Theorem	36
References and Additional Reading	37
Exercises	38
II Private-Key (Symmetric) Cryptography	
3 Private-Key Encryption	43
3.1 Computational Security	43
3.1.1 The Concrete Approach	44
3.1.2 The Asymptotic Approach	45
3.2 Defining Computationally Secure Encryption	52
3.2.1 The Basic Definition of Security	53
3.2.2 *Semantic Security	56
3.3 Constructing Secure Encryption Schemes	60
3.3.1 Pseudorandom Generators and Stream Ciphers	60
3.3.2 Proofs by Reduction	65
3.3.3 A Secure Fixed-Length Encryption Scheme	66

3.4	Stronger Security Notions	71
3.4.1	Security for Multiple Encryptions	71
3.4.2	Chosen-Plaintext Attacks and CPA-Security	73
3.5	Constructing CPA-Secure Encryption Schemes	77
3.5.1	Pseudorandom Functions and Block Ciphers	77
3.5.2	CPA-Secure Encryption from Pseudorandom Functions	82
3.6	Modes of Operation	86
3.6.1	Stream-Cipher Modes of Operation	86
3.6.2	Block-Cipher Modes of Operation	88
3.7	Chosen-Ciphertext Attacks	96
3.7.1	Defining CCA-Security	96
3.7.2	Padding-Oracle Attacks	98
	References and Additional Reading	101
	Exercises	102
4	Message Authentication Codes	107
4.1	Message Integrity	107
4.1.1	Secrecy vs. Integrity	107
4.1.2	Encryption vs. Message Authentication	108
4.2	Message Authentication Codes – Definitions	110
4.3	Constructing Secure Message Authentication Codes	116
4.3.1	A Fixed-Length MAC	116
4.3.2	Domain Extension for MACs	118
4.4	CBC-MAC	122
4.4.1	The Basic Construction	123
4.4.2	*Proof of Security	125
4.5	Authenticated Encryption	131
4.5.1	Definitions	131
4.5.2	Generic Constructions	132
4.5.3	Secure Communication Sessions	140
4.5.4	CCA-Secure Encryption	141
4.6	*Information-Theoretic MACs	142
4.6.1	Constructing Information-Theoretic MACs	143
4.6.2	Limitations on Information-Theoretic MACs	145
	References and Additional Reading	146
	Exercises	147
5	Hash Functions and Applications	153
5.1	Definitions	153
5.1.1	Collision Resistance	154
5.1.2	Weaker Notions of Security	156
5.2	Domain Extension: The Merkle–Damgård Transform	156
5.3	Message Authentication Using Hash Functions	158
5.3.1	Hash-and-MAC	159
5.3.2	HMAC	161

5.4	Generic Attacks on Hash Functions	164
5.4.1	Birthday Attacks for Finding Collisions	164
5.4.2	Small-Space Birthday Attacks	166
5.4.3	*Time/Space Tradeoffs for Inverting Functions	168
5.5	The Random-Oracle Model	174
5.5.1	The Random-Oracle Model in Detail	175
5.5.2	Is the Random-Oracle Methodology Sound?	179
5.6	Additional Applications of Hash Functions	182
5.6.1	Fingerprinting and Deduplication	182
5.6.2	Merkle Trees	183
5.6.3	Password Hashing	184
5.6.4	Key Derivation	186
5.6.5	Commitment Schemes	187
	References and Additional Reading	189
	Exercises	189
6	Practical Constructions of Symmetric-Key Primitives	193
6.1	Stream Ciphers	194
6.1.1	Linear-Feedback Shift Registers	195
6.1.2	Adding Nonlinearity	197
6.1.3	Trivium	198
6.1.4	RC4	199
6.2	Block Ciphers	202
6.2.1	Substitution-Permutation Networks	204
6.2.2	Feistel Networks	211
6.2.3	DES – The Data Encryption Standard	212
6.2.4	3DES: Increasing the Key Length of a Block Cipher	220
6.2.5	AES – The Advanced Encryption Standard	223
6.2.6	*Differential and Linear Cryptanalysis	225
6.3	Hash Functions	231
6.3.1	Hash Functions from Block Ciphers	232
6.3.2	MD5	234
6.3.3	SHA-0, SHA-1, and SHA-2	234
6.3.4	SHA-3 (Keccak)	235
	References and Additional Reading	236
	Exercises	237
7	*Theoretical Constructions of Symmetric-Key Primitives	241
7.1	One-Way Functions	242
7.1.1	Definitions	242
7.1.2	Candidate One-Way Functions	245
7.1.3	Hard-Core Predicates	246
7.2	From One-Way Functions to Pseudorandomness	248
7.3	Hard-Core Predicates from One-Way Functions	250
7.3.1	A Simple Case	250

7.3.2	A More Involved Case	251
7.3.3	The Full Proof	254
7.4	Constructing Pseudorandom Generators	257
7.4.1	Pseudorandom Generators with Minimal Expansion	258
7.4.2	Increasing the Expansion Factor	259
7.5	Constructing Pseudorandom Functions	265
7.6	Constructing (Strong) Pseudorandom Permutations	269
7.7	Assumptions for Private-Key Cryptography	273
7.8	Computational Indistinguishability	276
	References and Additional Reading	278
	Exercises	279

III Public-Key (Asymmetric) Cryptography

8 Number Theory and Cryptographic Hardness Assumptions 285

8.1	Preliminaries and Basic Group Theory	287
8.1.1	Primes and Divisibility	287
8.1.2	Modular Arithmetic	289
8.1.3	Groups	291
8.1.4	The Group \mathbb{Z}_N^*	295
8.1.5	*Isomorphisms and the Chinese Remainder Theorem	297
8.2	Primes, Factoring, and RSA	302
8.2.1	Generating Random Primes	303
8.2.2	*Primality Testing	306
8.2.3	The Factoring Assumption	311
8.2.4	The RSA Assumption	312
8.2.5	*Relating the RSA and Factoring Assumptions	314
8.3	Cryptographic Assumptions in Cyclic Groups	316
8.3.1	Cyclic Groups and Generators	316
8.3.2	The Discrete-Logarithm/Diffie–Hellman Assumptions	319
8.3.3	Working in (Subgroups of) \mathbb{Z}_p^*	322
8.3.4	Elliptic Curves	325
8.4	*Cryptographic Applications	332
8.4.1	One-Way Functions and Permutations	332
8.4.2	Constructing Collision-Resistant Hash Functions	335
	References and Additional Reading	337
	Exercises	338

9 *Algorithms for Factoring and Computing Discrete Logarithms 341

9.1	Algorithms for Factoring	342
9.1.1	Pollard’s $p - 1$ Algorithm	343
9.1.2	Pollard’s Rho Algorithm	344
9.1.3	The Quadratic Sieve Algorithm	345
9.2	Algorithms for Computing Discrete Logarithms	348

9.2.1	The Pohlig–Hellman Algorithm	350
9.2.2	The Baby-Step/Giant-Step Algorithm	352
9.2.3	Discrete Logarithms from Collisions	353
9.2.4	The Index Calculus Algorithm	354
9.3	Recommended Key Lengths	356
	References and Additional Reading	357
	Exercises	358
10	Key Management and the Public-Key Revolution	359
10.1	Key Distribution and Key Management	359
10.2	A Partial Solution: Key-Distribution Centers	361
10.3	Key Exchange and the Diffie–Hellman Protocol	363
10.4	The Public-Key Revolution	370
	References and Additional Reading	372
	Exercises	373
11	Public-Key Encryption	375
11.1	Public-Key Encryption – An Overview	375
11.2	Definitions	378
11.2.1	Security against Chosen-Plaintext Attacks	379
11.2.2	Multiple Encryptions	381
11.2.3	Security against Chosen-Ciphertext Attacks	387
11.3	Hybrid Encryption and the KEM/DEM Paradigm	389
11.3.1	CPA-Security	393
11.3.2	CCA-Security	398
11.4	CDH/DDH-Based Encryption	399
11.4.1	El Gamal Encryption	400
11.4.2	DDH-Based Key Encapsulation	404
11.4.3	*A CDH-Based KEM in the Random-Oracle Model	406
11.4.4	Chosen-Ciphertext Security and DHIES/ECIES	408
11.5	RSA Encryption	410
11.5.1	Plain RSA	410
11.5.2	Padded RSA and PKCS #1 v1.5	415
11.5.3	*CPA-Secure Encryption without Random Oracles	417
11.5.4	OAEP and RSA PKCS #1 v2.0	421
11.5.5	*A CCA-Secure KEM in the Random-Oracle Model	425
11.5.6	RSA Implementation Issues and Pitfalls	429
	References and Additional Reading	432
	Exercises	433
12	Digital Signature Schemes	439
12.1	Digital Signatures – An Overview	439
12.2	Definitions	441
12.3	The Hash-and-Sign Paradigm	443
12.4	RSA Signatures	444

12.4.1	Plain RSA	444
12.4.2	RSA-FDH and PKCS #1 v2.1	446
12.5	Signatures from the Discrete-Logarithm Problem	451
12.5.1	The Schnorr Signature Scheme	451
12.5.2	DSA and ECDSA	459
12.6	*Signatures from Hash Functions	461
12.6.1	Lamport’s Signature Scheme	461
12.6.2	Chain-Based Signatures	465
12.6.3	Tree-Based Signatures	468
12.7	*Certificates and Public-Key Infrastructures	473
12.8	Putting It All Together – SSL/TLS	479
12.9	*Signcryption	481
	References and Additional Reading	483
	Exercises	484
13	*Advanced Topics in Public-Key Encryption	487
13.1	Public-Key Encryption from Trapdoor Permutations	487
13.1.1	Trapdoor Permutations	488
13.1.2	Public-Key Encryption from Trapdoor Permutations	489
13.2	The Paillier Encryption Scheme	491
13.2.1	The Structure of $\mathbb{Z}_{N^2}^*$	492
13.2.2	The Paillier Encryption Scheme	494
13.2.3	Homomorphic Encryption	499
13.3	Secret Sharing and Threshold Encryption	501
13.3.1	Secret Sharing	501
13.3.2	Verifiable Secret Sharing	503
13.3.3	Threshold Encryption and Electronic Voting	505
13.4	The Goldwasser–Micali Encryption Scheme	507
13.4.1	Quadratic Residues Modulo a Prime	507
13.4.2	Quadratic Residues Modulo a Composite	510
13.4.3	The Quadratic Residuosity Assumption	514
13.4.4	The Goldwasser–Micali Encryption Scheme	515
13.5	The Rabin Encryption Scheme	518
13.5.1	Computing Modular Square Roots	518
13.5.2	A Trapdoor Permutation Based on Factoring	523
13.5.3	The Rabin Encryption Scheme	527
	References and Additional Reading	528
	Exercises	529
	Index of Common Notation	533

Appendix A Mathematical Background	537
A.1 Identities and Inequalities	537
A.2 Asymptotic Notation	537
A.3 Basic Probability	538
A.4 The “Birthday” Problem	542
A.5 *Finite Fields	544
Appendix B Basic Algorithmic Number Theory	547
B.1 Integer Arithmetic	549
B.1.1 Basic Operations	549
B.1.2 The Euclidean and Extended Euclidean Algorithms	550
B.2 Modular Arithmetic	552
B.2.1 Basic Operations	552
B.2.2 Computing Modular Inverses	552
B.2.3 Modular Exponentiation	553
B.2.4 *Montgomery Multiplication	556
B.2.5 Choosing a Uniform Group Element	557
B.3 *Finding a Generator of a Cyclic Group	559
B.3.1 Group-Theoretic Background	559
B.3.2 Efficient Algorithms	561
References and Additional Reading	562
Exercises	562
References	563
Index	577



Preface

The goal of our book remains the same as in the first edition: to present the basic paradigms and principles of modern cryptography to a general audience with a basic mathematics background. We have designed this book to serve as a textbook for undergraduate- or graduate-level courses in cryptography (in computer science, electrical engineering, or mathematics departments), as a general introduction suitable for self-study (especially for beginning graduate students), and as a reference for students, researchers, and practitioners.

There are numerous other cryptography textbooks available today, and the reader may rightly ask whether another book on the subject is needed. We would not have written this book—nor worked on revising it for the second edition—if the answer to that question were anything other than an unequivocal *yes*. What, in our opinion, distinguishes our book from other available books is that it provides a *rigorous* treatment of modern cryptography in an *accessible* manner appropriate for an introduction to the topic.

Our focus is on *modern* (post-1980s) cryptography, which is distinguished from classical cryptography by its emphasis on definitions, precise assumptions, and rigorous proofs of security. We briefly discuss each of these in turn (these principles are explored in greater detail in Chapter 1):

- **The central role of definitions:** A key intellectual contribution of modern cryptography has been the recognition that *formal definitions of security are an essential first step in the design of any cryptographic primitive or protocol*. The reason, in retrospect, is simple: if you don't know what it is you are trying to achieve, how can you hope to know when you have achieved it? As we will see in this book, cryptographic definitions of security are quite strong and—at first glance—may appear impossible to achieve. One of the most amazing aspects of cryptography is that efficient constructions satisfying such strong definitions can be proven to exist (under rather mild assumptions).
- **The importance of precise assumptions:** As will be explained in Chapters 2 and 3, many cryptographic constructions cannot currently be proven secure in an unconditional sense. Security often relies, instead, on some widely believed (though unproven) assumption(s). The modern cryptographic approach dictates that *any such assumption must be clearly stated and unambiguously defined*. This not only allows for objective evaluation of the assumption but, more importantly, enables rigorous proofs of security as described next.

- **The possibility of proofs of security:** The previous two principles serve as the basis for the idea that *cryptographic constructions can be proven secure* with respect to clearly stated definitions of security and relative to well-defined cryptographic assumptions. This concept is the essence of modern cryptography, and is what has transformed the field from an art to a science.

The importance of this idea cannot be overemphasized. Historically, cryptographic schemes were designed in a largely ad hoc fashion, and were deemed to be secure if the designers themselves could not find any attacks. In contrast, modern cryptography advocates the design of schemes with formal, mathematical proofs of security in well-defined models. Such schemes are *guaranteed* to be secure unless the underlying assumption is false (or the security definition did not appropriately model the real-world security concerns). By relying on long-standing assumptions (e.g., the assumption that “factoring is hard”), it is thus possible to obtain schemes that are extremely unlikely to be broken.

A unified approach. The above principles of modern cryptography are relevant not only to the “theory of cryptography” community. The importance of precise definitions is, by now, widely understood and appreciated by developers and security engineers who use cryptographic tools to build secure systems, and rigorous proofs of security have become one of the requirements for cryptographic schemes to be standardized.

Changes in the Second Edition

In preparing the second edition, we have made a conscious effort to integrate a more practical perspective (without sacrificing a rigorous approach). This is reflected in a number of changes and additions we have made:

- We have increased our coverage of *stream ciphers*, introducing them as a variant of pseudorandom generators in Section 3.3.1, discussing stream-cipher modes of operation in Section 3.6.1, and describing modern stream-cipher design principles and examples in Section 6.1.
- We have emphasized the importance of *authenticated encryption* (see Section 4.5) and have added a section on secure communication sessions.
- We have moved our treatment of hash functions into its own chapter (Chapter 5), have included some standard applications of cryptographic hash functions (Section 5.6), and have added a section on hash-function design principles and widely used constructions (Section 6.3). We have also improved our treatment of birthday attacks (covering small-space birthday attacks in Section 5.4.2) and have added a discussion of rainbow tables and time/space tradeoffs (Section 5.4.3).

- We have included several important attacks on implementations of cryptography that arise in practice, including chosen-plaintext attacks on chained-CBC encryption (Section 3.6.2), padding-oracle attacks on CBC-mode encryption (Section 3.7.2), and timing attacks on MAC verification (Section 4.2).
- After much deliberation, we have decided to introduce the random-oracle model much earlier in the book (Section 5.5). This allows us to give a proper, integrated treatment of standardized, widely used public-key encryption and signature schemes in later chapters, instead of relegating them to second-class status in a chapter at the end of the book.
- We have strengthened our coverage of elliptic-curve cryptography (Section 8.3.4) and have added a discussion of its impact on recommended key lengths (Section 9.3).
- In the chapter on public-key encryption, we introduce the KEM/DEM paradigm as a form of hybrid encryption (see Section 11.3). We also cover DHIES/ECIES in addition to the RSA PKCS #1 standards.
- In the chapter on digital signatures, we now describe the construction of signatures from identification schemes using the Fiat–Shamir transform, with the Schnorr signature scheme as a prototypical example. We have also improved our coverage of DSA/ECDSA. We include brief discussions of SSL/TLS and signcryption, both of which serve as culminations of everything covered up to that point.
- In the “advanced topics” chapter, we have amplified our treatment of homomorphic encryption, and have included sections on secret sharing and threshold encryption.

Beyond the above, we have also edited the entire book to make extensive corrections as well as smaller adjustments, including more worked examples, to improve the exposition. Several additional exercises have also been added.

Guide to Using This Book

This section is intended primarily for instructors seeking to adopt this book for their course, though the student picking up this book on his or her own may also find it a useful overview.

Required background. We have structured the book so that the only formal prerequisite is a course on discrete mathematics. Even here we rely on very little material: we assume familiarity with basic (discrete) probability and modular arithmetic. Students reading this book are also expected to have had some exposure to algorithms, mainly to be comfortable reading pseudocode and to be familiar with big- \mathcal{O} notation. Many of these concepts are reviewed in Appendix A and/or when first used in the book.

Notwithstanding the above, the book does use definitions, proofs, and abstract mathematical concepts, and therefore requires some mathematical maturity. In particular, the reader is assumed to have had some exposure to proofs at the college level, whether in an upper-level mathematics course or a course on discrete mathematics, algorithms, or computability theory.

Suggestions for course organization. The core material of this book, which we recommend should be covered in any introductory course on cryptography, consists of the following (in all cases, starred sections are excluded; more on this below):

- *Introduction and Classical Cryptography:* Chapters 1 and 2 discuss classical cryptography and set the stage for modern cryptography.
- *Private-Key (Symmetric) Cryptography:* Chapter 3 on private-key encryption, Chapter 4 on message authentication, and Chapter 5 on hash functions provide a thorough treatment of these topics.

We also highly recommend covering Section 6.2, which deals with block-cipher design; in our experience students really enjoy this material, and it makes the abstract ideas they have learned in previous chapters more concrete. Although we do consider this core material, it is not used in the rest of the book and so can be safely skipped if desired.

- *Public-Key (Asymmetric) Cryptography:* Chapter 8 gives a self-contained introduction to all the number theory needed for the remainder of the book. The material in Chapter 9 is not used subsequently; however, we do recommend at least covering Section 9.3 on recommended key lengths. The public-key revolution is described in Chapter 10. Ideally, all of Chapters 11 and 12 should be covered; those pressed for time can pick and choose appropriately.

We are typically able to cover most of the above in a one-semester (35-hour) undergraduate course (omitting some proofs and skipping some topics, as needed) or, with some changes to add more material on theoretical foundations, in the first three-quarters of a one-semester graduate course. Instructors with more time available can proceed at a more leisurely pace or incorporate additional topics, as discussed below.

Those wishing to cover additional material, in either a longer course or a faster-paced graduate course, will find that the book is structured to allow flexible incorporation of other topics as time permits (and depending on the interests of the instructor). Specifically, the starred (*) sections and chapters may be covered in any order, or skipped entirely, without affecting the overall flow of the book. We have taken care to ensure that none of the core material depends on any of the starred material and, for the most part, the starred sections do not depend on each other. (When they do, this dependence is explicitly noted.)

We suggest the following from among the starred topics for those wishing to give their course a particular flavor:

- *Theory*: A more theoretically inclined course could include material from Section 3.2.2 (semantic security); Chapter 7 (one-way functions and hard-core predicates, and constructing pseudorandom generators, functions, and permutations from one-way permutations); Section 8.4 (one-way functions and collision-resistant hash functions from number-theoretic assumptions); Section 11.5.3 (RSA encryption without random oracles); and Section 12.6 (signatures without random oracles).
- *Mathematics*: A course directed at students with a strong mathematics background—or being taught by someone who enjoys this aspect of cryptography—could incorporate Section 4.6 (information-theoretic MACs in finite fields); some of the more advanced number theory from Chapter 8 (e.g., the Chinese remainder theorem and the Miller–Rabin primality test); and all of Chapter 9.

In either case, a selection of advanced topics from Chapter 13 could also be included.

Feedback and Errata

Our goal in writing this book was to make modern cryptography accessible to a wide audience beyond the “theoretical computer science” community. We hope you will let us know if we have succeeded. The many enthusiastic emails we have received in response to our first edition have made the whole process of writing this book worthwhile.

We are always happy to receive feedback. We hope there are no errors or typos in the book; if you do find any, however, we would greatly appreciate it if you let us know. (A list of known errata will be maintained at <http://www.cs.umd.edu/~jkatz/imc.html>.) You can email your comments and errata to jkatz@cs.umd.edu and lindell@biu.ac.il; please put “Introduction to Modern Cryptography” in the subject line.

Acknowledgments

For the second edition: We are grateful to the many readers of the first edition who have sent us comments, suggestions, and corrections that helped to greatly improve the book. Discussions with Claude Crépeau, Bill Gasarch, Gene Itkis, Leonid Reyzin, Tom Shrimpton, and Salil Vadhan regarding the content and overall “philosophy” of the book were especially fruitful. We also thank Bar Alon, Gilad Asharov, Giuseppe Ateniese, Amir Azodi, Omer Berkman, Sergio de Biasi, Aurora Bristol, Richard Chang, Qingfeng Cheng, Kwan Tae Cho, Kylliah Clarkson, Ran Cohen, Nikolas Coukouma, Dana Dachman-Soled, Michael Fang, Michael Farcasin, Pooya Farshim, Marc Fischlin, Lance

Fortnow, Michael Fuhr, Bill Gasarch, Virgil Gligor, Carmit Hazay, Andreas Hübner, Karst Koymans, Eyal Kushilevitz, Steve Lai, Ugo Dal Lago, Armand Makowski, Tal Malkin, Steve Myers, Naveen Nathan, Ariel Nof, Eran Omri, Ruy de Queiroz, Eli Quiroz, Tal Rabin, Charlie Rackoff, Yona Raekow, Tzachy Reinman, Wei Ren, Ben Riva, Volker Roth, Christian Schaffner, Joachim Schipper, Dominique Schröder, Randy Shull, Nigel Smart, Christoph Sprenger, Aravind Srinivasan, John Steinberger, Aishwarya Thiruvengadam, Dave Tuller, Poorvi Vora, Avishai Yanai, Rupeng Yang, Arkady Yerukhimovich, Dae Hyun Yum, Hila Zarosim, and Konstantin Ziegler for their helpful corrections to the first edition and/or early drafts of the second edition.

For the first edition: We thank Zoe Bermant for producing the figures; David Wagner for answering questions related to block ciphers and their cryptanalysis; and Salil Vadhan and Alon Rosen for experimenting with an early version of our text in an introductory course at Harvard University and for providing us with valuable feedback. We would also like to extend our gratitude to those who read and commented on earlier drafts of this book and to those who sent us corrections: Adam Bender, Chiu-Yuen Koo, Yair Dombb, Michael Fuhr, William Glenn, S. Dov Gordon, Carmit Hazay, Eyal Kushilevitz, Avivit Levy, Matthew Mah, Ryan Murphy, Steve Myers, Martin Paraskevov, Eli Quiroz, Jason Rogers, Rui Xue, Dicky Yan, Arkady Yerukhimovich, and Hila Zarosim. We are extremely grateful to all those who encouraged us to write this book and agreed with us that a book of this sort is badly needed.

Finally, we thank our wives and children for all their support and understanding during the many hours, days, months, and now years we have spent on this project.