
Contents

Preface	xv
I Introduction and Classical Cryptography	
1 Introduction	1
1.1 Cryptography and Modern Cryptography	1
1.2 The Setting of Private-Key Encryption	2
1.3 Historical Ciphers and Their Cryptanalysis	6
1.4 Principles of Modern Cryptography	14
1.4.1 Principle 1 – Formal Definitions	15
1.4.2 Principle 2 – Precise Assumptions	18
1.4.3 Principle 3 – Proofs of Security	20
1.4.4 Provable Security and Real-World Security	20
References and Additional Reading	21
Exercises	21
2 Perfectly Secret Encryption	23
2.1 Definitions	24
2.2 The One-Time Pad	31
2.3 Limitations of Perfect Secrecy	33
2.4 *Shannon’s Theorem	34
References and Additional Reading	36
Exercises	36
II Private-Key (Symmetric) Cryptography	41
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	51
3.2.1 The Basic Definition of Security (EAV-Security)	52
3.2.2 *Semantic Security	56
3.3 Constructing an EAV-Secure Encryption Scheme	60
3.3.1 Pseudorandom Generators	60
3.3.2 Proofs by Reduction	64
3.3.3 EAV-Security from a Pseudorandom Generator	65

3.4	Stronger Security Notions	70
3.4.1	Security for Multiple Encryptions	70
3.4.2	Chosen-Plaintext Attacks and CPA-Security	72
3.4.3	CPA-Security for Multiple Encryptions	74
3.5	Constructing a CPA-Secure Encryption Scheme	75
3.5.1	Pseudorandom Functions and Permutations	76
3.5.2	CPA-Security from a Pseudorandom Function	80
3.6	Modes of Operation and Encryption in Practice	84
3.6.1	Stream Ciphers	85
3.6.2	Stream-Cipher Modes of Operation	87
3.6.3	Block Ciphers and Block-Cipher Modes of Operation	88
3.6.4	*Nonce-Based Encryption	96
	References and Additional Reading	99
	Exercises	99
4	Message Authentication Codes	105
4.1	Message Integrity	105
4.1.1	Secrecy vs. Integrity	105
4.1.2	Encryption vs. Message Authentication	106
4.2	Message Authentication Codes (MACs) – Definitions	108
4.3	Constructing Secure Message Authentication Codes	114
4.3.1	A Fixed-Length MAC	114
4.3.2	Domain Extension for MACs	116
4.4	CBC-MAC	120
4.4.1	The Basic Construction	120
4.4.2	*Proof of Security	123
4.5	GMAC and Poly1305	128
4.5.1	MACs from Difference-Universal Functions	128
4.5.2	Instantiations	131
4.6	*Information-Theoretic MACs	133
4.6.1	One-Time MACs from Strongly Universal Functions	134
4.6.2	One-Time MACs from Difference-Universal Functions	137
4.6.3	Limitations on Information-Theoretic MACs	139
	References and Additional Reading	140
	Exercises	140
5	CCA-Security and Authenticated Encryption	145
5.1	Chosen-Ciphertext Attacks and CCA-Security	145
5.1.1	Padding-Oracle Attacks	146
5.1.2	Defining CCA-Security	149
5.2	Authenticated Encryption	151
5.2.1	Defining Authenticated Encryption	151
5.2.2	CCA Security vs. Authenticated Encryption	153
5.3	Authenticated Encryption Schemes	154
5.3.1	Generic Constructions	154

5.3.2	Standardized Schemes	161
5.4	Secure Communication Sessions	162
	References and Additional Reading	164
	Exercises	164
6	Hash Functions and Applications	167
6.1	Definitions	167
6.1.1	Collision Resistance	168
6.1.2	Weaker Notions of Security	170
6.2	Domain Extension: The Merkle–Damgård Transform	170
6.3	Message Authentication Using Hash Functions	172
6.3.1	Hash-and-MAC	172
6.3.2	HMAC	175
6.4	Generic Attacks on Hash Functions	177
6.4.1	Birthday Attacks for Finding Collisions	178
6.4.2	Small-Space Birthday Attacks	179
6.4.3	*Time/Space Tradeoffs for Inverting Hash Functions	182
6.5	The Random-Oracle Model	187
6.5.1	The Random-Oracle Model in Detail	188
6.5.2	Is the Random-Oracle Methodology Sound?	192
6.6	Additional Applications of Hash Functions	195
6.6.1	Fingerprinting and Deduplication	195
6.6.2	Merkle Trees	196
6.6.3	Password Hashing	198
6.6.4	Key Derivation	199
6.6.5	Commitment Schemes	200
	References and Additional Reading	202
	Exercises	203
7	Practical Constructions of Symmetric-Key Primitives	207
7.1	Stream Ciphers	208
7.1.1	Linear-Feedback Shift Registers	209
7.1.2	Adding Nonlinearity	211
7.1.3	Trivium	212
7.1.4	RC4	213
7.1.5	ChaCha20	216
7.2	Block Ciphers	217
7.2.1	Substitution-Permutation Networks	219
7.2.2	Feistel Networks	226
7.2.3	DES – The Data Encryption Standard	228
7.2.4	3DES: Increasing the Key Length of a Block Cipher	235
7.2.5	AES – The Advanced Encryption Standard	238
7.2.6	*Differential and Linear Cryptanalysis	240
7.3	Compression Functions and Hash Functions	246
7.3.1	Compression Functions from Block Ciphers	246

7.3.2	MD5, SHA-1, and SHA-2	249
7.3.3	The Sponge Construction and SHA-3 (Keccak)	250
	References and Additional Reading	254
	Exercises	255
8	*Theoretical Constructions of Symmetric-Key Primitives	261
8.1	One-Way Functions	262
8.1.1	Definitions	262
8.1.2	Candidate One-Way Functions	265
8.1.3	Hard-Core Predicates	266
8.2	From One-Way Functions to Pseudorandomness	267
8.3	Hard-Core Predicates from One-Way Functions	269
8.3.1	A Simple Case	270
8.3.2	A More Involved Case	270
8.3.3	The Full Proof	274
8.4	Constructing Pseudorandom Generators	277
8.4.1	Pseudorandom Generators with Minimal Expansion	277
8.4.2	Increasing the Expansion Factor	279
8.5	Constructing Pseudorandom Functions	284
8.6	Constructing (Strong) Pseudorandom Permutations	289
8.7	Assumptions for Private-Key Cryptography	293
8.8	Computational Indistinguishability	296
	References and Additional Reading	298
	Exercises	299
III	Public-Key (Asymmetric) Cryptography	303
9	Number Theory and Cryptographic Hardness Assumptions	305
9.1	Preliminaries and Basic Group Theory	306
9.1.1	Primes and Divisibility	307
9.1.2	Modular Arithmetic	309
9.1.3	Groups	311
9.1.4	The Group \mathbb{Z}_N^*	315
9.1.5	*Isomorphisms and the Chinese Remainder Theorem	317
9.2	Primes, Factoring, and RSA	322
9.2.1	Generating Random Primes	323
9.2.2	*Primality Testing	325
9.2.3	The Factoring Assumption	331
9.2.4	The RSA Assumption	331
9.2.5	*Relating the Factoring and RSA Assumptions	334
9.3	Cryptographic Assumptions in Cyclic Groups	336
9.3.1	Cyclic Groups and Generators	336
9.3.2	The Discrete-Logarithm/Diffie–Hellman Assumptions	339
9.3.3	Working in (Subgroups of) \mathbb{Z}_p^*	342
9.3.4	Elliptic Curves	345

9.4	*Cryptographic Applications	354
9.4.1	One-Way Functions and Permutations	355
9.4.2	Collision-Resistant Hash Functions	357
	References and Additional Reading	359
	Exercises	360
10	*Algorithms for Factoring and Computing Discrete Logarithms	365
10.1	Algorithms for Factoring	366
10.1.1	Pollard's $p - 1$ Algorithm	367
10.1.2	Pollard's Rho Algorithm	368
10.1.3	The Quadratic Sieve Algorithm	369
10.2	Generic Algorithms for Computing Discrete Logarithms	372
10.2.1	The Pohlig–Hellman Algorithm	374
10.2.2	The Baby-Step/Giant-Step Algorithm	376
10.2.3	Discrete Logarithms from Collisions	377
10.3	Index Calculus: Computing Discrete Logarithms in \mathbb{Z}_p^*	378
10.4	Recommended Key Lengths	380
	References and Additional Reading	381
	Exercises	382
11	Key Management and the Public-Key Revolution	385
11.1	Key Distribution and Key Management	385
11.2	A Partial Solution: Key-Distribution Centers	387
11.3	Key Exchange and the Diffie–Hellman Protocol	389
11.4	The Public-Key Revolution	396
	References and Additional Reading	398
	Exercises	399
12	Public-Key Encryption	401
12.1	Public-Key Encryption – An Overview	401
12.2	Definitions	404
12.2.1	Security against Chosen-Plaintext Attacks	405
12.2.2	Multiple Encryptions	407
12.2.3	Security against Chosen-Ciphertext Attacks	412
12.3	Hybrid Encryption and the KEM/DEM Paradigm	415
12.3.1	CPA-Security	419
12.3.2	CCA-Security	424
12.4	CDH/DDH-Based Encryption	425
12.4.1	El Gamal Encryption	426
12.4.2	DDH-Based Key Encapsulation	430
12.4.3	*A CDH-Based KEM in the Random-Oracle Model	432
12.4.4	*Chosen-Ciphertext Security and DHIES/ECIES	434
12.5	RSA-Based Encryption	436
12.5.1	Plain RSA Encryption	436

12.5.2	Padded RSA and PKCS #1 v1.5	441
12.5.3	*CPA-Secure Encryption without Random Oracles	443
12.5.4	OAEP and PKCS #1 v2	447
12.5.5	*A CCA-Secure KEM in the Random-Oracle Model	451
12.5.6	RSA Implementation Issues and Pitfalls	455
	References and Additional Reading	458
	Exercises	459
13	Digital Signature Schemes	463
13.1	Digital Signatures – An Overview	463
13.2	Definitions	465
13.3	The Hash-and-Sign Paradigm	467
13.4	RSA-Based Signatures	468
13.4.1	Plain RSA Signatures	468
13.4.2	RSA-FDH and PKCS #1 Standards	470
13.5	Signatures from the Discrete-Logarithm Problem	475
13.5.1	Identification Schemes and Signatures	475
13.5.2	The Schnorr Identification/Signature Schemes	480
13.5.3	DSA and ECDSA	483
13.6	Certificates and Public-Key Infrastructures	485
13.7	Putting It All Together – TLS	491
13.8	*Signcryption	493
	References and Additional Reading	495
	Exercises	495
14	*Post-Quantum Cryptography	499
14.1	Post-Quantum Symmetric-Key Cryptography	500
14.1.1	Grover’s Algorithm and Symmetric-Key Lengths	500
14.1.2	Collision-Finding Algorithms and Hash Functions	501
14.2	Shor’s Algorithm and its Impact on Cryptography	502
14.3	Post-Quantum Public-Key Encryption	504
14.4	Post-Quantum Signatures	509
14.4.1	Lamport’s Signature Scheme	510
14.4.2	Chain-Based Signatures	513
14.4.3	Tree-Based Signatures	517
	References and Additional Reading	522
	Exercises	523
15	*Advanced Topics in Public-Key Encryption	525
15.1	Public-Key Encryption from Trapdoor Permutations	525
15.1.1	Trapdoor Permutations	526
15.1.2	Public-Key Encryption from Trapdoor Permutations	527
15.2	The Paillier Encryption Scheme	529
15.2.1	The Structure of $\mathbb{Z}_{N^2}^*$	530
15.2.2	The Paillier Encryption Scheme	532

15.2.3	Homomorphic Encryption	537
15.3	Secret Sharing and Threshold Encryption	539
15.3.1	Secret Sharing	539
15.3.2	Verifiable Secret Sharing	541
15.3.3	Threshold Encryption and Electronic Voting	543
15.4	The Goldwasser–Micali Encryption Scheme	545
15.4.1	Quadratic Residues Modulo a Prime	545
15.4.2	Quadratic Residues Modulo a Composite	548
15.4.3	The Quadratic Residuosity Assumption	552
15.4.4	The Goldwasser–Micali Encryption Scheme	553
15.5	The Rabin Encryption Scheme	556
15.5.1	Computing Modular Square Roots	556
15.5.2	A Trapdoor Permutation Based on Factoring	561
15.5.3	The Rabin Encryption Scheme	565
	References and Additional Reading	566
	Exercises	567
Index of Common Notation		571
Appendix A Mathematical Background		575
A.1	Identities and Inequalities	575
A.2	Asymptotic Notation	575
A.3	Basic Probability	576
A.4	The “Birthday” Problem	581
A.5	*Finite Fields	584
Appendix B Basic Algorithmic Number Theory		587
B.1	Integer Arithmetic	589
B.1.1	Basic Operations	589
B.1.2	The Euclidean and Extended Euclidean Algorithms	590
B.2	Modular Arithmetic	591
B.2.1	Basic Operations	592
B.2.2	Computing Modular Inverses	592
B.2.3	Modular Exponentiation	593
B.2.4	*Montgomery Multiplication	595
B.2.5	Choosing a Uniform Group Element	597
B.3	*Finding a Generator of a Cyclic Group	599
B.3.1	Group-Theoretic Background	599
B.3.2	Efficient Algorithms	601
	References and Additional Reading	602
	Exercises	602
References		603
Index		619

Preface

The goal of our book remains the same as in the first edition: to present the core 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 and third editions—if the answer to that question were anything other than an unequivocal *yes*. What, in our opinion, distinguishes our book from others is that it provides a *rigorous* treatment of modern cryptography in an *accessible* and *introductory* manner.

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 unconditionally. Security, instead, generally relies on some widely believed (though unproven) assumption(s). The modern cryptographic approach dictates that *any such assumptions must be clearly stated and unambiguously defined*. This not only allows for objective evaluation of the assumptions 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 heuristic 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 (with respect to a certain security definition) unless the underlying assumption is false. By relying on long-standing assumptions, 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 Third Edition

In preparing the third edition, we have continued to integrate a more practical perspective without sacrificing a rigorous approach. This is reflected in a number of changes and additions as compared to the second edition:

- We have divided our treatment of symmetric-key encryption into two parts: Chapter 3 deals with security against “passive” attacks (i.e., CPA-security), while Chapter 5 addresses “active” attacks (i.e., CCA-security and authenticated encryption). Besides breaking up what was previously a long chapter, this also allows us to introduce message authentication codes before discussing active attacks against encryption schemes.
- With an eye toward symmetric-key schemes used in practice, we have improved our coverage of stream ciphers and stream-cipher modes of operation (Sections 3.6.1 and 3.6.2); added a treatment of nonce-based encryption (Section 3.6.4); and incorporated material about standardized schemes such as GMAC and Poly1305 (Section 4.5) as well as GCM, CCM, and ChaCha20-Poly1305 (Section 5.3.2).
- With similar motivation, we have added sections on the ChaCha20 stream cipher and SHA-3 to Chapter 7. As part of our discussion about SHA-3, we also describe the sponge construction.

- We have further increased our coverage of elliptic-curve cryptography (Section 9.3.4), including a discussion of elliptic curves used in practice.
- Our treatment of TLS in Section 13.7 has been updated to reflect the latest version (TLS 1.3).
- Reflecting recent trends, we have added a chapter (Chapter 14) describing the impact of quantum computers on cryptography, and providing examples of “post-quantum” encryption and signature schemes.

For those currently using the first edition of our book, as well as for reference, we also summarize the changes/additions we have already made in the second edition (all of which remain here):

- We have increased our coverage of *stream ciphers*, including stream-cipher modes of operation as well as stream-cipher design principles and examples of stream ciphers used in practice.
- We have emphasized the importance of authenticated encryption and secure communication sessions in Sections 5.2–5.4.
- We have moved our treatment of hash functions into its own chapter (Chapter 6), and have added a section on hash-function design principles and widely used constructions (Section 7.3). We have also improved our treatment of generic attacks on hash functions, including a discussion of rainbow tables (Section 6.4.3).
- We have included several important attacks on cryptographic *implementations* that arise in practice, including chosen-plaintext attacks on chained-CBC encryption (Section 3.6.3), timing attacks on MAC verification (Section 4.2), and padding-oracle attacks on CBC-mode encryption (Section 5.1.1).
- After much deliberation, we have decided to introduce the random-oracle model earlier in the book (Section 6.5). This has several benefits, including allowing for an integrated treatment of standardized public-key encryption and signature schemes in Chapters 12 and 13.
- We have strengthened our coverage of elliptic-curve cryptography (Section 9.3.4) and have added a discussion of its impact on recommended key lengths (Section 10.4).
- In the chapter on public-key encryption, we introduce the KEM/DEM paradigm as a form of hybrid encryption (see Section 12.3). 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 material covered up to that point.

- In the “advanced topics” chapter, we have amplified our treatment of homomorphic encryption, and have added 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 the only formal prerequisite is a course on discrete mathematics. Even here we rely on very little: we only 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, 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 3–5 provide a thorough treatment of private-key encryption and message authentication, and Chapter 6 covers hash functions and their applications. (Section 6.6 could be skipped if that material will not be used later.)

We also highly recommend covering at least part of Chapter 7, which deals with symmetric-key primitives used in practice; 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 remainder of the book and so can be safely skipped if desired.

- *Public-Key Cryptography*: Chapter 9 gives a self-contained introduction to all the number theory needed for the remainder of the book. The material in The public-key revolution, including Diffie–Hellman key exchange, is described in Chapter 11. Chapters 12 and 13 go into detail about public-key encryption and digital signatures; those pressed for time can pick and choose what to cover appropriately.

We are typically able to cover most of the above in a one-semester (35-hour) undergraduate or Masters-level 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 PhD-level 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 (i.e., unstarred) 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 8 (one-way functions and hard-core predicates, and constructing pseudorandom generators, functions, and permutations from one-way permutations); Section 9.4 (one-way functions and collision-resistant hash functions from number-theoretic assumptions); Section 12.5.3 (RSA encryption without random oracles); and Section 15.3 (cryptographic protocols).
- *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 9 (e.g., the Chinese remainder theorem, the Miller–Rabin primality test, and more on elliptic curves); and all of Chapter 10 (algorithms for factoring and computing discrete logarithms).

In either case, a selection of advanced public-key schemes from Chapters 14 and 15 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 and second editions 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. You can email your comments and errata to jkatz2@gmail.com and lindell@biu.ac.il; please put “Introduction to Modern Cryptography” in the subject line. A list of known errata will be maintained at <http://www.cs.umd.edu/~jkatz/imc.html>.

Acknowledgments

We continue to be grateful to all those who have sent us comments, suggestions, and corrections for the book. We would like to thank, in particular, Jack Aaron, Rounak Agarwal, Ionut Ambrosie, Dan Bernstein, Jeremiah Blocki, David Cash, Claude Crépeau, Dana Dachman-Soled, Daniel Escudero, Pooya Farshim, Rolf Haenni, Imededdine Jerbi, Ali El Kaafarani, Zach Kissel, Angélique Faye Loe, Wilde Luo, Tal Malkin, Alejandro Mardones, Kurt Pan, Greg Plaxton, Kyle Andrew Porter, Christian Schaffner, Jim Tallent, Hanh Tang, Markus Triska, and Rui Xue for their feedback on the second edition.

Finally, we thank our wives and children for all their support during the now over a decade(!) we have spent working on this project.