

Information Security and Cryptography

Series Editors

David Basin
Kenny Paterson

Advisory Board

Michael Backes
Gilles Barthe
Ronald Cramer
Ivan Damgård
Andrew D. Gordon
Joshua D. Guttman
Christopher Kruegel
Ueli Maurer
Tatsuaki Okamoto
Adrian Perrig
Bart Preneel

More information about this series at <http://www.springer.com/series/4752>

Nigel P. Smart

Cryptography Made Simple

 Springer

Nigel P. Smart
University of Bristol
Bristol, UK

ISSN 1619-7100 ISSN 2197-845X (electronic)
Information Security and Cryptography
ISBN 978-3-319-21935-6 ISBN 978-3-319-21936-3 (eBook)
DOI 10.1007/978-3-319-21936-3

Library of Congress Control Number: 2015955608

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

This is a reworking of my earlier book “Cryptography: An Introduction” which has been available online for over a decade. In the intervening years there have been major advances and changes in the subject which have led me to revisit much of the material in this book. In the main the book remains the same, in that it tries to present a non-rigorous treatment of modern cryptography, which is itself a highly rigorous area of computer science/mathematics. Thus the book acts as a stepping stone between more “traditional” courses which are taught to undergraduates around the world, and the more advanced rigorous courses taught in graduate school.

The motivation for such a bridging book is that, in my view, the traditional courses (which deal with basic RSA encryption and signatures, and perhaps AES) are not a suitable starting point. They do not emphasize the importance of what it means for a system to be secure; and are often introduced into a curriculum as a means of demonstrating the applicability of mathematical theory as opposed to developing the material as a subject in its own right. However, most undergraduates could not cope with a full-on rigorous treatment from the start. After all one first needs to get a grasp of basic ideas before one can start building up a theoretical edifice.

The main differences between this version and the Third Edition of “Cryptography: An Introduction” is in the ordering of material. Now security definitions are made central to the discussion of modern cryptography, and all discussions of attacks and weaknesses are related back to these definitions. We have found this to be a good way of presenting the material over the last few years in Bristol; hence the reordering. In addition many topics have been updated, and explanations improved. I have also made a number of the diagrams more pleasing to the eye.

Cryptography courses are now taught at all major universities; sometimes these are taught in the context of a Mathematics degree, sometimes in the context of a Computer Science degree, and sometimes in the context of an Electrical Engineering degree. Indeed, a single course often needs to meet the requirements of all three types of students, plus maybe some from other subjects who are taking the course as an “open unit”. The backgrounds and needs of these students are different; some will require a quick overview of the algorithms currently in use, whilst others will want an introduction to current research directions. Hence, there seems to be a need for a textbook which starts from a low level and builds confidence in students until they are able to read the texts mentioned at the end of this Preface.

The background I assume is what one could expect of a third or fourth year undergraduate in computer science. One can assume that such students have already met the basics of discrete mathematics (modular arithmetic) and a little probability. In addition, they will have at some point done (but probably forgotten) elementary calculus. Not that one needs calculus for cryptography, but the ability to happily deal with equations and symbols is certainly helpful. Apart from that I introduce everything needed from scratch. For those students who wish to dig into the mathematics a little more, or who need some further reading, I have provided an appendix which covers most of the basic algebra and notation needed to cope with modern cryptosystems.

It is quite common for computer science courses not to include much of complexity theory or formal methods. Many such courses are based more on software engineering and applications of computer science to areas such as graphics, vision or artificial intelligence. The main goal of such courses is to train students for the workplace rather than to delve into the theoretical aspects of

the subject. Hence, I have introduced what parts of theoretical computer science I need, as and when required.

I am not mathematically rigorous at all steps, given the target audience, but aim to give a flavour of the mathematics involved. For example I often only give proof outlines, or may not worry about the success probabilities of many of the reductions. I try to give enough of the gory details to demonstrate why a protocol or primitive has been designed in a certain way. Readers wishing for a more in-depth study of the various points covered or a more mathematically rigorous coverage should consult one of the textbooks or papers in the Further Reading sections at the end of each chapter.

On the other hand we use the terminology of groups and finite fields from the outset. This is for two reasons. Firstly, it equips students with the vocabulary to read the latest research papers, and hence enables students to carry on their studies at the research level. Secondly, students who do not progress to study cryptography at the postgraduate level will find that to understand practical issues in the “real world”, such as API descriptions and standards documents, a knowledge of this terminology is crucial. We have taken this approach with our students in Bristol, who do not have any prior exposure to this form of mathematics, and find that it works well as long as abstract terminology is introduced alongside real-world concrete examples and motivation.

I have always found that when reading protocols and systems for the first time the hardest part is to work out what is public information and which information one is trying to keep private. This is particularly true when one meets a public key encryption algorithm for the first time, or one is deciphering a substitution cipher. Hence I have continued with the colour coding from the earlier book. Generally speaking items in **red** are secret and should never be divulged to anyone. Items in **blue** are public information and are known to everyone, or are known to the party one is currently pretending to be.

For example, suppose one is trying to break a system and recover some secret message m ; suppose the attacker computes some quantity b . Here the **red** refers to the quantity the attacker does not know and **blue** refers to the quantity the attacker does know. If one is then able to write down, after some algebra,

$$b = \dots = m,$$

then it is clear something is wrong with our cryptosystem. The attacker has found out something he should not. This colour coding will be used at all places where it adds something to the discussion. In other situations, where the context is clear or all data is meant to be secret, I do not bother with the colours.

To aid self-study each chapter is structured as follows:

- A list of items the chapter will cover, so you know what you will be told about.
- The actual chapter contents.
- A summary of what the chapter contains. This will be in the form of revision notes: if you wish to commit anything to memory it should be these facts.
- Further Reading. Each chapter contains a list of a few books or papers from which further information can be obtained. Such pointers are mainly to material which you should be able to tackle given that you have read the prior chapter.

There are no references made to other work in this book; it is a textbook and I did not want to break the flow with references to this, that and the other. Therefore, you should not assume that ANY of the results in this book are my own; in fact NONE are my own. Those who wish to obtain pointers to the literature should consult one of the books mentioned in the Further Reading sections.

The book is clearly too large for a single course on cryptography; this gives the instructor using the book a large range of possible threads through the topics. For a traditional cryptography course within a Mathematics department I would recommend Chapters 1, 2, 3, 7, 11, 12, 13, 14, 15, 16

and 17. For a course in a Computer Science department I would recommend Chapters 1, 11, 12, 13, 14, 15 and 16, followed by a selection from 18, 19, 20, 21 and 22. In any course I *strongly* recommend the material in Chapter 11 should be covered. This is to enable students to progress to further study, or to be able to deal with the notions which occur when using cryptography in the real world. The other chapters in this book provide additional supplementary material on historical matters, implementation aspects, or act as introductions to topics found in the recent literature.

Special thanks go to the following people (whether academics, students or industrialists) for providing input over the years on the various versions of the material: Nils Anderson, Endre Bangerter, Guy Barwell, David Bernhard, Dan Bernstein, Ian Blake, Colin Boyd, Sergiu Bursuc, Jiun-Ming Chen, Joan Daemen, Ivan Damgård, Gareth Davies, Reza Rezaeian Farashahi, Ed Geraghty, Florian Hess, Nick Howgrave-Graham, Ellen Jochemsz, Thomas Johansson, Georgios Kafanas, Parimal Kumar, Jake Longo Galea, Eugene Luks, Vadim Lyubashevsky, David McCann, Bruce McIntosh, John Malone-Lee, Wenbo Mao, Dan Martin, John Merriman, Phong Nguyen, Emmanuela Orsini, Dan Page, Christopher Peikert, Joop van de Pol, David Rankin, Vincent Rijmen, Ron Rivest, Michal Rybar, Berry Schoenmakers, Tom Shrimpton, Martijn Stam, Ryan Stanley, Damien Stehle, Edlyn Teske, Susan Thomson, Frederik Vercauteren, Bogdan Warinschi, Carolyn Whitnall, Steve Williams and Marcin Wójcik.

Nigel Smart
University of Bristol

Further Reading

After finishing this book if you want to know more technical details then I would suggest the following books:

A.J. Menezes, P. van Oorschot and S.A. Vanstone. *The Handbook of Applied Cryptography*. CRC Press, 1997.

J. Katz and Y. Lindell. *Introduction to Modern Cryptography: Principles and Protocols*. CRC Press, 2007.

Contents

Preface	v
Part 1. Mathematical Background	1
Chapter 1. Modular Arithmetic, Groups, Finite Fields and Probability	3
1.1. Modular Arithmetic	3
1.2. Finite Fields	8
1.3. Basic Algorithms	11
1.4. Probability	21
1.5. Big Numbers	24
Chapter 2. Primality Testing and Factoring	27
2.1. Prime Numbers	27
2.2. The Factoring and Factoring-Related Problems	32
2.3. Basic Factoring Algorithms	38
2.4. Modern Factoring Algorithms	42
2.5. Number Field Sieve	44
Chapter 3. Discrete Logarithms	51
3.1. The DLP, DHP and DDH Problems	51
3.2. Pohlig–Hellman	54
3.3. Baby-Step/Giant-Step Method	57
3.4. Pollard-Type Methods	59
3.5. Sub-exponential Methods for Finite Fields	64
Chapter 4. Elliptic Curves	67
4.1. Introduction	67
4.2. The Group Law	69
4.3. Elliptic Curves over Finite Fields	72
4.4. Projective Coordinates	74
4.5. Point Compression	75
4.6. Choosing an Elliptic Curve	77
Chapter 5. Lattices	79
5.1. Lattices and Lattice Reduction	79
5.2. “Hard” Lattice Problems	85
5.3. q -ary Lattices	89
5.4. Coppersmith’s Theorem	90
Chapter 6. Implementation Issues	95
6.1. Introduction	95
6.2. Exponentiation Algorithms	95
6.3. Special Exponentiation Methods	99

6.4. Multi-precision Arithmetic	101
6.5. Finite Field Arithmetic	107
Part 2. Historical Ciphers	117
Chapter 7. Historical Ciphers	119
7.1. Introduction	119
7.2. Shift Cipher	120
7.3. Substitution Cipher	123
7.4. Vigenère Cipher	126
7.5. A Permutation Cipher	131
Chapter 8. The Enigma Machine	133
8.1. Introduction	133
8.2. An Equation for the Enigma	136
8.3. Determining the Plugboard Given the Rotor Settings	137
8.4. Double Encryption of Message Keys	140
8.5. Determining the Internal Rotor Wirings	141
8.6. Determining the Day Settings	147
8.7. The Germans Make It Harder	148
8.8. Known Plaintext Attack and the Bombes	150
8.9. Ciphertext Only Attack	158
Chapter 9. Information-Theoretic Security	163
9.1. Introduction	163
9.2. Probability and Ciphers	164
9.3. Entropy	169
9.4. Spurious Keys and Unicity Distance	173
Chapter 10. Historical Stream Ciphers	179
10.1. Introduction to Symmetric Ciphers	179
10.2. Stream Cipher Basics	181
10.3. The Lorenz Cipher	182
10.4. Breaking the Lorenz Cipher's Wheels	188
10.5. Breaking a Lorenz Cipher Message	192
Part 3. Modern Cryptography Basics	195
Chapter 11. Defining Security	197
11.1. Introduction	197
11.2. Pseudo-random Functions and Permutations	197
11.3. One-Way Functions and Trapdoor One-Way Functions	201
11.4. Public Key Cryptography	202
11.5. Security of Encryption	203
11.6. Other Notions of Security	209
11.7. Authentication: Security of Signatures and MACs	215
11.8. Bit Security	219
11.9. Computational Models: The Random Oracle Model	221
Chapter 12. Modern Stream Ciphers	225
12.1. Stream Ciphers from Pseudo-random Functions	225
12.2. Linear Feedback Shift Registers	227

12.3.	Combining LFSRs	233
12.4.	RC4	238
Chapter 13.	Block Ciphers and Modes of Operation	241
13.1.	Introduction to Block Ciphers	241
13.2.	Feistel Ciphers and DES	244
13.3.	AES	250
13.4.	Modes of Operation	254
13.5.	Obtaining Chosen Ciphertext Security	266
Chapter 14.	Hash Functions, Message Authentication Codes and Key Derivation Functions	271
14.1.	Collision Resistance	271
14.2.	Padding	275
14.3.	The Merkle–Damgård Construction	276
14.4.	The MD-4 Family	278
14.5.	HMAC	282
14.6.	Merkle–Damgård-Based Key Derivation Function	284
14.7.	MACs and KDFs Based on Block Ciphers	285
14.8.	The Sponge Construction and SHA-3	288
Chapter 15.	The “Naive” RSA Algorithm	295
15.1.	“Naive” RSA Encryption	295
15.2.	“Naive” RSA Signatures	299
15.3.	The Security of RSA	301
15.4.	Some Lattice-Based Attacks on RSA	305
15.5.	Partial Key Exposure Attacks on RSA	309
15.6.	Fault Analysis	310
Chapter 16.	Public Key Encryption and Signature Algorithms	313
16.1.	Passively Secure Public Key Encryption Schemes	313
16.2.	Random Oracle Model, OAEP and the Fujisaki–Okamoto Transform	319
16.3.	Hybrid Ciphers	324
16.4.	Constructing KEMs	329
16.5.	Secure Digital Signatures	333
16.6.	Schemes Avoiding Random Oracles	342
Chapter 17.	Cryptography Based on Really Hard Problems	349
17.1.	Cryptography and Complexity Theory	349
17.2.	Knapsack-Based Cryptosystems	353
17.3.	Worst-Case to Average-Case Reductions	356
17.4.	Learning With Errors (LWE)	360
Chapter 18.	Certificates, Key Transport and Key Agreement	369
18.1.	Introduction	369
18.2.	Certificates and Certificate Authorities	371
18.3.	Fresh Ephemeral Symmetric Keys from Static Symmetric Keys	375
18.4.	Fresh Ephemeral Symmetric Keys from Static Public Keys	382
18.5.	The Symbolic Method of Protocol Analysis	388
18.6.	The Game-Based Method of Protocol Analysis	392
Part 4.	Advanced Protocols	401
Chapter 19.	Secret Sharing Schemes	403

19.1.	Access Structures	403
19.2.	General Secret Sharing	405
19.3.	Reed–Solomon Codes	407
19.4.	Shamir Secret Sharing	412
19.5.	Application: Shared RSA Signature Generation	414
Chapter 20.	Commitments and Oblivious Transfer	417
20.1.	Introduction	417
20.2.	Commitment Schemes	417
20.3.	Oblivious Transfer	421
Chapter 21.	Zero-Knowledge Proofs	425
21.1.	Showing a Graph Isomorphism in Zero-Knowledge	425
21.2.	Zero-Knowledge and \mathcal{NP}	428
21.3.	Sigma Protocols	429
21.4.	An Electronic Voting System	436
Chapter 22.	Secure Multi-party Computation	439
22.1.	Introduction	439
22.2.	The Two-Party Case	441
22.3.	The Multi-party Case: Honest-but-Curious Adversaries	445
22.4.	The Multi-party Case: Malicious Adversaries	448
Appendix		451
Basic Mathematical Terminology		453
A.1.	Sets	453
A.2.	Relations	453
A.3.	Functions	455
A.4.	Permutations	456
A.5.	Operations	459
A.6.	Groups	461
A.7.	Rings	468
A.8.	Fields	469
A.9.	Vector Spaces	470
Index		475