

89103200028



b89103200028a

UNDERGRADUATE TEXTS IN MATHEMATICS

This self-contained introduction to modern cryptography emphasizes the mathematics behind the theory of public key cryptosystems and digital signature schemes. The book focuses on these key topics while developing the mathematical tools needed for the construction and security analysis of diverse cryptosystems. Only basic linear algebra is required of the reader; techniques from algebra, number theory, and probability are introduced and developed as required.

The book covers a variety of topics that are considered central to mathematical cryptography. Key topics include:

- classical cryptographic constructions, such as Diffie-Hellmann key exchange, discrete logarithm-based cryptosystems, the RSA cryptosystem, and digital signatures;
- fundamental mathematical tools for cryptography, including primality testing, factorization algorithms, probability theory, information theory, and collision algorithms;
- an in-depth treatment of important recent cryptographic innovations, such as elliptic curves, elliptic curve and pairing-based cryptography, lattices, lattice-based cryptography, and the NTRU cryptosystem.

This book is an ideal introduction for mathematics and computer science students to the mathematical foundations of modern cryptography. The book includes an extensive bibliography and index; supplementary materials are available online.

ISBN 978-0-387-77993-5



9 780387 779935

springer.com

Hoffstein · Pipher
Silverman

UTM



An Introduction to Mathematical Cryptography

QA
268
H64
2008

MATH

Jeffrey Hoffstein
Jill Pipher
Joseph H. Silverman

UNDERGRADUATE TEXTS IN MATHEMATICS

An Introduction to Mathematical Cryptography



Springer

Jeffrey Hoffstein
Jill Pipher
Joseph H. Silverman

An Introduction to Mathematical Cryptography

 Springer

Jeffrey Hoffstein
Department of Mathematics
Brown University
151 Thayer St.
Providence, RI 02912
USA
jhoff@math.brown.edu

Jill Pipher
Department of Mathematics
Brown University
151 Thayer St.
Providence, RI 02912
USA
jpipher@math.brown.edu

Joseph H. Silverman
Department of Mathematics
Brown University
151 Thayer St.
Providence, RI 02912
USA
jhs@math.brown.edu

Editorial Board

S. Axler
Mathematics Department
San Francisco State University
San Francisco, CA 94132
USA
axler@sfsu.edu

K.A. Ribet
Department of Mathematics
University of California
at Berkeley
Berkeley, CA 94720
USA
ribet@math.berkeley.edu

ISBN: 978-0-387-77993-5

e-ISBN: 978-0-387-77994-2

DOI: 10.1007/978-0-387-77994-2

Library of Congress Control Number: 2008923038

Mathematics Subject Classification (2000): 94A60, 11T71, 14G50, 68P25

© 2008 Springer Science+Business Media, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

General Library System
University of Wisconsin - Madison
728 State Street
Madison, WI 53706-1494
U.S.A.

Contents

Preface	v
Introduction	xi
1 An Introduction to Cryptography	1
1.1 Simple substitution ciphers	1
1.2 Divisibility and greatest common divisors	10
1.3 Modular arithmetic	19
1.4 Prime numbers, unique factorization, and finite fields	26
1.5 Powers and primitive roots in finite fields	29
1.6 Cryptography before the computer age	34
1.7 Symmetric and asymmetric ciphers	36
Exercises	47
2 Discrete Logarithms and Diffie–Hellman	59
2.1 The birth of public key cryptography	59
2.2 The discrete logarithm problem	62
2.3 Diffie–Hellman key exchange	65
2.4 The ElGamal public key cryptosystem	68
2.5 An overview of the theory of groups	72
2.6 How hard is the discrete logarithm problem?	75
2.7 A collision algorithm for the DLP	79
2.8 The Chinese remainder theorem	81
2.9 The Pohlig–Hellman algorithm	86
2.10 Rings, quotients, polynomials, and finite fields	92
Exercises	105
3 Integer Factorization and RSA	113
3.1 Euler’s formula and roots modulo pq	113
3.2 The RSA public key cryptosystem	119
3.3 Implementation and security issues	122
3.4 Primality testing	124
3.5 Pollard’s $p - 1$ factorization algorithm	133

3.6	Factorization via difference of squares	137
3.7	Smooth numbers and sieves	146
3.8	The index calculus and discrete logarithms	162
3.9	Quadratic residues and quadratic reciprocity	165
3.10	Probabilistic encryption	172
	Exercises	176
4	Combinatorics, Probability, and Information Theory	189
4.1	Basic principles of counting	190
4.2	The Vigenère cipher	196
4.3	Probability theory	210
4.4	Collision algorithms and meet-in-the-middle attacks	227
4.5	Pollard's ρ method	234
4.6	Information theory	243
4.7	Complexity Theory and $\mathcal{P}$ versus $\mathcal{NP}$	258
	Exercises	262
5	Elliptic Curves and Cryptography	279
5.1	Elliptic curves	279
5.2	Elliptic curves over finite fields	286
5.3	The elliptic curve discrete logarithm problem	290
5.4	Elliptic curve cryptography	296
5.5	The evolution of public key cryptography	301
5.6	Lenstra's elliptic curve factorization algorithm	303
5.7	Elliptic curves over $\mathbb{F}_2$ and over $\mathbb{F}_{2^k}$	308
5.8	Bilinear pairings on elliptic curves	315
5.9	The Weil pairing over fields of prime power order	325
5.10	Applications of the Weil pairing	334
	Exercises	339
6	Lattices and Cryptography	349
6.1	A congruential public key cryptosystem	349
6.2	Subset-sum problems and knapsack cryptosystems	352
6.3	A brief review of vector spaces	359
6.4	Lattices: Basic definitions and properties	363
6.5	Short vectors in lattices	370
6.6	Babai's algorithm	379
6.7	Cryptosystems based on hard lattice problems	383
6.8	The GGH public key cryptosystem	384
6.9	Convolution polynomial rings	387
6.10	The NTRU public key cryptosystem	392
6.11	NTRU as a lattice cryptosystem	400
6.12	Lattice reduction algorithms	403
6.13	Applications of LLL to cryptanalysis	418
	Exercises	422

7	Digital Signatures	437
7.1	What is a digital signature?	437
7.2	RSA digital signatures	440
7.3	ElGamal digital signatures and DSA	442
7.4	GGH lattice-based digital signatures	447
7.5	NTRU digital signatures	450
	Exercises	458
8	Additional Topics in Cryptography	465
8.1	Hash functions	466
8.2	Random numbers and pseudorandom number generators	468
8.3	Zero-knowledge proofs	470
8.4	Secret sharing schemes	473
8.5	Identification schemes	474
8.6	Padding schemes and the random oracle model	476
8.7	Building protocols from cryptographic primitives	479
8.8	Hyperelliptic curve cryptography	480
8.9	Quantum computing	483
8.10	Modern symmetric cryptosystems: DES and AES	485
	List of Notation	489
	References	493
	Index	501

It turns out that this transformation can be achieved in polynomial time. Shor then applies the quantum Fourier transform to a certain superposition of states and measures the resulting system. The key computation shows that the probability of seeing state $|c\rangle$ is relatively large if there exists a rational number $\frac{d}{r} \in \mathbb{Q}$ satisfying

$$\left| \frac{c}{q} - \frac{d}{r} \right| < \frac{1}{2q}.$$

(Recall that r is the order of a .) Using the continued fraction expansion of the known rational number $\frac{c}{q}$, it is not hard to determine the fraction $\frac{d}{r}$ in lowest terms, since $q > n^2$.

There remains only the “minor” challenge of building a functioning quantum computer. Research in this field has focused on the issue of decoherence, which involves controlling the errors in quantum computation introduced by the interaction of the computer with its environment. There is already a vast literature on quantum computing and quantum computers, reflecting to some extent the large amount of government funding that has been allocated to the subject. One place to start gathering resources about quantum computers is the website for NIST’s Quantum Information Program at qubit.nist.gov.

Finally, we would be remiss if we did not mention the theory of *quantum cryptography*. The idea is to use quantum-mechanical principles such as the Heisenberg uncertainty principle or the entanglement of quantum states to perform a completely secure key exchange. In particular, if Eve attempts to read either Bob’s or Alice’s transmission, then quantum theory says that she must alter the data, so Bob and Alice will know that their communication has been compromised.

8.10 Modern symmetric cryptosystems: DES and AES

In Section 1.7.1 we gave an abstract formulation of symmetric ciphers, and in Section 1.7.4 we described several elementary examples. Not surprisingly, none of the examples in Section 1.7.4 is secure. Modern symmetric ciphers such as the *Data Encryption Standard* (DES) and the *Advanced Encryption Standard* (AES) are based on ad hoc mixing operations, rather than on intractable mathematical problems used by asymmetric ciphers. The reason that DES and AES and other symmetric ciphers are used in practice is that they are much faster than asymmetric ciphers. Thus if Alice wants to send Bob a long message, she first uses an asymmetric cipher such as RSA to send Bob a key for a symmetric cipher, and then she uses a symmetric cipher such as DES or AES to send the actual data.

DES was created by a team of cryptographers at IBM in the early 1970s, and with some modifications suggested by the United States National Security Agency (NSA), it was officially adopted in 1977 as a government standard

suitable for use in commercial applications. (See [83].) DES uses a 56-bit private key and encrypts blocks of 64 bits at a time. Most of DES's mixing operations are linear, with the only nonlinear component being the use of eight *S-boxes* (substitution boxes). Each S-box is a look-up table in which six input bits are replaced by four output bits. Figure 8.1 illustrates one of the S-boxes used by DES.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Figure 8.1: The first of eight S-boxes used by DES

Here is how an S-box is used. The input is a list of six bits, say

$$\text{Input} = \beta_1\beta_2\beta_3\beta_4\beta_5\beta_6.$$

First use the 2-bit binary number $\beta_1\beta_6$, which is a number between 0 and 3, to choose the row of the S-box, and then use the 4-bit binary number $\beta_2\beta_3\beta_4\beta_5$, which is a number between 0 and 15, to choose the column of the S-box. The output is the entry of the S-box for the chosen row and column. This entry, which is between 0 and 15, is converted into a 4-bit number.

For example, suppose that the input string is '110010'. Binary '10' is 2, so we use row 2, and binary '1001' is 9, so we use column 9. The entry of the S-box in Figure 8.1 for row 2 and column 9 is 12, which we convert to binary '1100'.

The S-boxes were designed to prevent various sorts of attacks, including especially an attack called differential cryptanalysis, which was known to IBM and the NSA in the 1970s, but published only after its rediscovery by Biham and Shamir in the 1980s. Differential cryptanalysis and other non-brute-force attacks are somewhat impractical because they require knowledge of a large number ($> 2^{40}$) of plaintext/ciphertext pairs.

A more serious flaw of DES is its comparatively short 56-bit key. As computer hardware became increasingly fast and inexpensive and computing power more distributed in the 1990s, it became feasible to break DES by a brute-force search of all possible keys, either using many machines over the Internet or building a dedicated DES cracking machine. Comparatively inexpensive machines now exist that are capable of breaking a DES key in less than a week.

One solution to this problem, which has been widely adopted, is to use DES multiple times. There are a number of different versions of *Triple DES*, the simplest of which is to simply encrypt the plaintext three times using three different keys. Thus if we write $\text{DES}(k, m)$ for the DES encryption of the message m using the key k , then one version of triple DES is

$$\text{TDES}(k_1, k_2, k_3, m) = \text{DES}(k_3, \text{DES}(k_2, \text{DES}(k_1, m))).$$

A variation replaces the middle DES encryption by a DES decryption; this has the effect that setting $k_1 = k_2 = k_3 = k$ yields ordinary DES encryption. Another variation, used by the electronics payment industry, takes $k_1 = k_3$, which reduces key size at the cost of some security reduction. Finally, since three DES encryptions triple the encryption time, another version called DES-X uses a single DES encryption combined with initial and final XOR operations with two 64-bit keys. Thus DES-X looks like

$$\text{DESX}(k_1, k_2, k_3, m) = k_3 \text{ xor } \text{DES}(k_2, m \text{ xor } k_1).$$

Although DES and its variants were widely deployed, it suffers from short and inflexible key and block sizes. Further, although DES is fast when implemented in specialized hardware, it is comparatively slow in software. So in 1997 the United States National Institute of Standards (NIST) organized an open competition to choose a replacement for DES. There were many submissions, and after several years of analysis and several international conferences devoted to the selection process, NIST announced in 2000 that the Rijndael cipher, invented by the Belgian cryptographers J. Daemen and V. Rijmen, had been chosen as AES. Since that time AES has been widely adopted, although variants of DES are still in use.¹³

AES is a block cipher in which the plaintext–ciphertext blocks are 128 bits in length and the key size may be 128, 192, or 256 bits. AES is similar to DES in that it encrypts and decrypts by repeating a basic operation several times. In the case of AES, there are 10, 12, or 14 rounds depending on the size of the key. AES is also similar to DES in that it uses an S-box to provide the all-important nonlinearity to the encryption process. However, AES's S-box is constructed using the operation of taking multiplicative inverses in the field $\mathbb{F}_{2^8}$ with 2^8 elements. (See Section 2.10.4 for a discussion of finite fields with a prime power number of elements.) Many of AES's basic operations use 128 bit blocks, which are broken up into 16 bytes. Each byte consists of 8 bits and is treated as an element of the field $\mathbb{F}_{2^8}$. Then various operations, including inversion, are performed in $\mathbb{F}_{2^8}$. The details of AES are too complicated to give here, but they are designed to be very fast when implemented in either software or hardware. The interested reader will find a full description in the official NIST publication FIPS PUB 197 [82] and in many other sources.

¹³All of the AES finalists (MARS, RC6, Rijndael, Serpent, Twofish) were believed to be secure, and none was clearly superior in all aspects. So the choice of Rijndael was based on its balance of flexibility, ease of implementation, and speed in both hardware and software.