"...the best introduction to cryptography I've ever seen....The book the National Security Agency wanted never to be published...."
—*Wired* magazine

MORE THAN 160,000 COPIES SOLD

SECOND EDITION

APPLIED CRYPTOGRAPHY

Protocols, Algorithms, and Source Code in C

BRUCE SCHNEIER

WILEY

SCHNEIER

APPLIED CRYPTOGRAPHY

SECOND
EDITION

WILEY

from reviews of the first edition of

APPLIED CRYPTOGRAPHY

Protocols, Algorithms, and Source Code in C

". . . the definitive text on the subject. . . ."
—*Software Development Magazine*

". . . good reading for anyone interested in cryptography."
—*BYTE*

"This book should be on the shelf of any computer professional involved in the use or implementation of cryptography."
—*IEEE Software*

". . . dazzling . . . fascinating. . . . This book *absolutely must* be on your bookshelf . . ."
—*PC Techniques*

". . . comprehensive . . . an encyclopedic work . . ."
—*The Cryptogram*

". . . a fantastic book on cryptography today. It belongs in the library of anyone interested in cryptography or anyone who deals with information security and cryptographic systems."
—*Computers & Security*

"An encyclopedic survey . . . could well have been subtitled 'The Joy of Encrypting' . . . a useful addition to the library of any active or would-be security practitioner."
—*Cryptologia*

". . . encyclopedic . . . readable . . . well-informed . . . picks up where Dorothy Denning's classic *Cryptography and Data Security* left off a dozen years ago. . . . This book would be a bargain at twice the price."
—*;login:*

"This is a marvelous resource—the best book on cryptography and its application available today."
—Dorothy Denning
Georgetown University

". . . Schneier's book is an indispensable reference and resource. . . . I recommend it highly."
—Martin Hellman
Stanford University

Errata

A list of the errors found in this book along with corresponding corrections is updated periodically. For the most recent electronic version, send email to:

schneier@counterpane.com

For the most recent printed version, send a stamped, self-addressed envelope to:

AC Corrections
Counterpane Systems
101 E. Minnekaka Parkway
Minneapolis, MN 55419

Readers are encouraged to distribute electronic or printed versions of this list to other readers of this book.

APPLIED CRYPTOGRAPHY, SECOND EDITION

PROTOCOLS, ALGORITHMS, AND SOURCE CODE IN C

BRUCE SCHNEIER

John Wiley & Sons, Inc.
New York • Chichester • Brisbane • Toronto • Singapore

Publisher: Katherine Schowalter
Editor: Phil Sutherland
Assistant Editor: Allison Roarty
Managing Editor: Robert Aronds
Text Design & Composition: North Market Street Graphics

This text is printed on acid-free paper.

Published by John Wiley & Sons, Inc.

In no event will the publisher or author be liable for any consequential, incidental, or indirect damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising from the use or inability to use the protocols and algorithms in this book, even if the publisher or author has been advised of the possibility of such damages.

Some of the protocols and algorithms in this book are protected by patents and copyrights. It is the responsibility of the reader to obtain all necessary patent and copyright licenses before implementing in software any protocol or algorithm in this book. This book does not contain an exhaustive list of all applicable patents and copyrights.

Some of the protocols and algorithms in this book are regulated under the United States Department of State International Traffic in Arms Regulations. It is the responsibility of the reader to obtain all necessary export licenses before implementing in software for export any protocol or algorithm in this book.

Library of Congress Cataloging-in-Publication Data:

Schneier, Bruce
Applied Cryptography Second Edition : protocols, algorithms, and source code in C / Bruce Schneier.
p. cm.
Includes bibliographical references (p. 675).
ISBN 0-471-12845-7 (cloth : acid-free paper). — ISBN 0-471-11709-9 (paper : acid-free paper)
1. Computer security. 2. Telecommunication—Security measures. 3. Cryptography. I. Title.
QA76.9.A25S35 1996
005.8'2—dc20 95-12398
CIP

Printed in the United States of America
20

Contents in Brief

Contents

PART I CRYPTOGRAPHIC PROTOCOLS

PART II CRYPTOGRAPHIC TECHNIQUES

PART III CRYPTOGRAPHIC ALGORITHMS

PART IV THE REAL WORLD

PART V SOURCE CODE

Foreword
By Whitfield Diffie

The literature of cryptography has a curious history. Secrecy, of course, has always played a central role, but until the First World War, important developments appeared in print in a more or less timely fashion and the field moved forward in much the same way as other specialized disciplines. As late as 1918, one of the most influential cryptanalytic papers of the twentieth century, William F. Friedman's monograph *The Index of Coincidence and Its Applications in Cryptography,* appeared as a research report of the private Riverbank Laboratories [577]. And this, despite the fact that the work had been done as part of the war effort. In the same year Edward H. Hebern of Oakland, California filed the first patent for a rotor machine [710], the device destined to be a mainstay of military cryptography for nearly 50 years.

After the First World War, however, things began to change. U.S. Army and Navy organizations, working entirely in secret, began to make fundamental advances in cryptography. During the thirties and forties a few basic papers did appear in the open literature and several treatises on the subject were published, but the latter were farther and farther behind the state of the art. By the end of the war the transition was complete. With one notable exception, the public literature had died. That exception was Claude Shannon's paper "The Communication Theory of Secrecy Systems," which appeared in the *Bell System Technical Journal* in 1949 [1432]. It was similar to Friedman's 1918 paper, in that it grew out of wartime work of Shannon's. After the Second World War ended it was declassified, possibly by mistake.

From 1949 until 1967 the cryptographic literature was barren. In that year a different sort of contribution appeared: David Kahn's history, *The Codebreakers* [794]. It didn't contain any new technical ideas, but it did contain a remarkably complete history of what had gone before, including mention of some things that the government still considered secret. The significance of *The Codebreakers* lay not just in its remarkable scope, but also in the fact that it enjoyed good sales and made tens of thousands of people, who had never given the matter a moment's thought, aware of cryptography. A trickle of new cryptographic papers began to be written.

At about the same time, Horst Feistel, who had earlier worked on identification friend or foe devices for the Air Force, took his lifelong passion for cryptography to the IBM Watson Laboratory in Yorktown Heights, New York. There, he began development of what was to become the U.S. Data Encryption Standard; by the early 1970s several technical reports on this subject by Feistel and his colleagues had been made public by IBM [1482,1484,552].

This was the situation when I entered the field in late 1972. The cryptographic literature wasn't abundant, but what there was included some very shiny nuggets.

Cryptology presents a difficulty not found in normal academic disciplines: the need for the proper interaction of cryptography and cryptanalysis. This arises out of the fact that in the absence of real communications requirements, it is easy to propose a system that appears unbreakable. Many academic designs are so complex that the would-be cryptanalyst doesn't know where to start; exposing flaws in these designs is far harder than designing them in the first place. The result is that the competitive process, which is one strong motivation in academic research, cannot take hold.

When Martin Hellman and I proposed public-key cryptography in 1975 [496], one of the indirect aspects of our contribution was to introduce a problem that does not even appear easy to solve. Now an aspiring cryptosystem designer could produce something that would be recognized as clever—something that did more than just turn meaningful text into nonsense. The result has been a spectacular increase in the number of people working in cryptography, the number of meetings held, and the number of books and papers published.

In my acceptance speech for the Donald E. Fink award—given for the best expository paper to appear in an IEEE journal—which I received jointly with Hellman in 1980, I told the audience that in writing "Privacy and Authentication," I had an experience that I suspected was rare even among the prominent scholars who populate the IEEE awards ceremony: I had written the paper I had wanted to study, but could not find, when I first became seriously interested in cryptography. Had I been able to go to the Stanford bookstore and pick up a modern cryptography text, I would probably have learned about the field years earlier. But the only things available in the fall of 1972 were a few classic papers and some obscure technical reports.

The contemporary researcher has no such problem. The problem now is choosing where to start among the thousands of papers and dozens of books. The contemporary researcher, yes, but what about the contemporary programmer or engineer who merely wants to use cryptography? Where does that person turn? Until now, it has been necessary to spend long hours hunting out and then studying the research literature before being able to design the sort of cryptographic utilities glibly described in popular articles.

This is the gap that Bruce Schneier's *Applied Cryptography* has come to fill. Beginning with the objectives of communication security and elementary examples of programs used to achieve these objectives, Schneier gives us a panoramic view of the fruits of 20 years of public research. The title says it all; from the mundane objective of having a secure conversation the very first time you call someone to the possibilities of digital money and cryptographically secure elections, this is where you'll find it.

Not satisfied that the book was about the real world merely because it went all the way down to the code, Schneier has included an account of the world in which cryptography is developed and applied, and discusses entities ranging from the International Association for Cryptologic Research to the NSA.

When public interest in cryptography was just emerging in the late seventies and early eighties, the National Security Agency (NSA), America's official cryptographic organ, made several attempts to quash it. The first was a letter from a long-time NSA employee allegedly, avowedly, and apparently acting on his own. The letter was sent to the IEEE and warned that the publication of cryptographic material was a violation of the International Traffic in Arms Regulations (ITAR). This viewpoint turned out not even to be supported by the regulations themselves—which contained an explicit exemption for published material—but gave both the public practice of cryptography and the 1977 Information Theory Workshop lots of unexpected publicity.

A more serious attempt occurred in 1980, when the NSA funded the American Council on Education to examine the issue with a view to persuading Congress to give it legal control of publications in the field of cryptography. The results fell far short of NSA's ambitions and resulted in a program of voluntary review of cryptographic papers; researchers were requested to ask the NSA's opinion on whether disclosure of results would adversely affect the national interest before publication.

As the eighties progressed, pressure focused more on the practice than the study of cryptography. Existing laws gave the NSA the power, through the Department of State, to regulate the export of cryptographic equipment. As business became more and more international and the American fraction of the world market declined, the pressure to have a single product in both domestic and offshore markets increased. Such single products were subject to export control and thus the NSA acquired substantial influence not only over what was exported, but also over what was sold in the United States.

As this is written, a new challenge confronts the public practice of cryptography. The government has augmented the widely published and available Data Encryption Standard, with a secret algorithm implemented in tamper-resistant chips. These chips will incorporate a codified mechanism of government monitoring. The negative aspects of this "key-escrow" program range from a potentially disastrous impact on personal privacy to the high cost of having to add hardware to products that had previously encrypted in software. So far key escrow products are enjoying less than stellar sales and the scheme has attracted widespread negative comment, especially from the independent cryptographers. Some people, however, see more future in programming than politicking and have redoubled their efforts to provide the world with strong cryptography that is accessible to public scrutiny.

A sharp step back from the notion that export control law could supersede the First Amendment seemed to have been taken in 1980 when the *Federal Register* announcement of a revision to ITAR included the statement: ". . . provision has been added to make it clear that the regulation of the export of technical data does not purport to interfere with the First Amendment rights of individuals." But the fact that tension between the First Amendment and the export control laws has not gone away should be evident from statements at a conference held by RSA Data Security. NSA's representative from the export control office expressed the opinion that people who published cryptographic programs were "in a grey area" with respect to the law. If that is so, it is a grey area on which the first edition of this book has shed some light. Export applications for the book itself have been granted, with acknowledgement that published material lay beyond the authority of the Munitions Control Board. Applications to export the enclosed programs on disk, however, have been denied.

The shift in the NSA's strategy, from attempting to control cryptographic research to tightening its grip on the development and deployment of cryptographic products, is presumably due to its realization that all the great cryptographic papers in the world do not protect a single bit of traffic. Sitting on the shelf, this volume may be able to do no better than the books and papers that preceded it, but sitting next to a workstation, where a programmer is writing cryptographic code, it just may.

Whitfield Diffie
Mountain View, CA

Preface

There are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files. This book is about the latter.

If I take a letter, lock it in a safe, hide the safe somewhere in New York, then tell you to read the letter, that's not security. That's obscurity. On the other hand, if I take a letter and lock it in a safe, and then give you the safe along with the design specifications of the safe and a hundred identical safes with their combinations so that you and the world's best safecrackers can study the locking mechanism—and you still can't open the safe and read the letter—that's security.

For many years, this sort of cryptography was the exclusive domain of the military. The United States' National Security Agency (NSA), and its counterparts in the former Soviet Union, England, France, Israel, and elsewhere, have spent billions of dollars in the very serious game of securing their own communications while trying to break everyone else's. Private individuals, with far less expertise and budget, have been powerless to protect their own privacy against these governments.

During the last 20 years, public academic research in cryptography has exploded. While classical cryptography has been long used by ordinary citizens, computer cryptography was the exclusive domain of the world's militaries since World War II. Today, state-of-the-art computer cryptography is practiced outside the secured walls of the military agencies. The layperson can now employ security practices that can protect against the most powerful of adversaries—security that may protect against military agencies for years to come.

Do average people really need this kind of security? Yes. They may be planning a political campaign, discussing taxes, or having an illicit affair. They may be designing a new product, discussing a marketing strategy, or planning a hostile business takeover. Or they may be living in a country that does not respect the rights of privacy of its citizens. They may be doing something that they feel shouldn't be illegal, but is. For whatever reason, the data and communications are personal, private, and no one else's business.

This book is being published in a tumultuous time. In 1994, the Clinton administration approved the Escrowed Encryption Standard (including the Clipper chip and Fortezza card) and signed the Digital Telephony bill into law. Both of these initiatives try to ensure the government's ability to conduct electronic surveillance.

Some dangerously Orwellian assumptions are at work here: that the government has the right to listen to private communications, and that there is something wrong with a private citizen trying to keep a secret from the government. Law enforcement has always been able to conduct court-authorized surveillance if possible, but this is the first time that the people have been forced to take active measures to *make themselves available* for surveillance. These initiatives are not simply government proposals in some obscure area; they are preemptive and unilateral attempts to usurp powers that previously belonged to the people.

Clipper and Digital Telephony do not protect privacy; they force individuals to unconditionally trust that the government will respect their privacy. The same law enforcement authorities who illegally tapped Martin Luther King Jr.'s phones can easily tap a phone protected with Clipper. In the recent past, local police authorities have either been charged criminally or sued civilly in numerous jurisdictions—Maryland, Connecticut, Vermont, Georgia, Missouri, and Nevada—for conducting illegal wiretaps. It's a poor idea to deploy a technology that could some day facilitate a police state.

The lesson here is that it is insufficient to protect ourselves with laws; we need to protect ourselves with mathematics. Encryption is too important to be left solely to governments.

This book gives you the tools you need to protect your own privacy; cryptography products may be declared illegal, but the information will never be.

HOW TO READ THIS BOOK

I wrote *Applied Cryptography* to be both a lively introduction to the field of cryptography and a comprehensive reference. I have tried to keep the text readable without sacrificing accuracy. This book is not intended to be a mathematical text. Although I have not deliberately given any false information, I do play fast and loose with theory. For those interested in formalism, there are copious references to the academic literature.

Chapter 1 introduces cryptography, defines many terms, and briefly discusses precomputer cryptography.

Chapters 2 through 6 (Part I) describe cryptographic protocols: what people can do with cryptography. The protocols range from the simple (sending encrypted messages from one person to another) to the complex (flipping a coin over the telephone) to the esoteric (secure and anonymous digital money exchange). Some of these protocols are obvious; others are almost amazing. Cryptography can solve a lot of problems that most people never realized it could.

Chapters 7 through 10 (Part II) discuss cryptographic techniques. All four chapters in this section are important for even the most basic uses of cryptography. Chapters 7 and 8 are about keys: how long a key should be in order to be secure, how to generate keys, how to store keys, how to dispose of keys, and so on. Key management is the hardest part of cryptography and often the Achilles' heel of an otherwise secure system. Chapter 9 discusses different ways of using cryptographic algorithms, and Chapter 10 gives the odds and ends of algorithms: how to choose, implement, and use algorithms.

Chapters 11 through 23 (Part III) list algorithms. Chapter 11 provides the mathematical background. This chapter is only required if you are interested in public-key algorithms. If you just want to implement DES (or something similar), you can skip ahead. Chapter 12 discusses DES: the algorithm, its history, its security, and some variants. Chapters 13, 14, and 15 discuss other block algorithms; if you want something more secure than DES, skip to the section on IDEA and triple-DES. If you want to read about a bunch of algorithms, some of which may be more secure than DES, read the whole chapter. Chapters 16 and 17 discuss stream algorithms. Chapter 18 focuses on one-way hash functions; MD5 and SHA are the most common, although I discuss many more. Chapter 19 discusses public-key encryption algorithms, Chapter 20 discusses public-key digital signature algorithms, Chapter 21 discusses public-key identification algorithms, and Chapter 22 discusses public-key key exchange algorithms. The important algorithms are RSA, DSA, Fiat-Shamir, and Diffie-Hellman, respectively. Chapter 23 has more esoteric public-key algorithms and protocols; the math in this chapter is quite complicated, so wear your seat belt.

Chapters 24 and 25 (Part IV) turn to the real world of cryptography. Chapter 24 discusses some of the current implementations of these algorithms and protocols, while Chapter 25 touches on some of the political issues surrounding cryptography. These chapters are by no means intended to be comprehensive.

Also included are source code listings for 10 algorithms discussed in Part III. I was unable to include all the code I wanted to due to space limitations, and cryptographic source code cannot otherwise be exported. (Amazingly enough, the State Department allowed export of the first edition of this book with source code, but denied export for a computer disk with the exact same source code on it. Go figure.) An associated source code disk set includes much more source code than I could fit in this book; it is probably the largest collection of cryptographic source code outside a military institution. I can only send source code disks to U.S. and Canadian citizens living in the U.S. and Canada, but hopefully that will change someday. If you are interested in implementing or playing with the cryptographic algorithms in this book, get the disk. See the last page of the book for details.

One criticism of this book is that its encyclopedic nature takes away from its readability. This is true, but I wanted to provide a single reference for those who might come across an algorithm in the academic literature or in a product. For those who are more interested in a tutorial, I apologize. A lot is being done in the field; this is the first time so much of it has been gathered between two covers. Even so, space considerations forced me to leave many things out. I covered topics that I felt were important, practical, or interesting. If I couldn't cover a topic in depth, I gave references to articles and papers that did.

I have done my best to hunt down and eradicate all errors in this book, but many have assured me that it is an impossible task. Certainly, the second edition has far fewer errors than the first. An errata listing is available from me and will be periodically posted to the Usenet newsgroup sci.crypt. If any reader finds an error, please let me know. I'll send the first person to find each error in the book a free copy of the source code disk.

Acknowledgments

The list of people who had a hand in this book may seem unending, but all are worthy of mention. I would like to thank Don Alvarez, Ross Anderson, Dave Balenson, Karl Barrus, Steve Bellovin, Dan Bernstein, Eli Biham, Joan Boyar, Karen Cooper, Whit Diffie, Joan Feigenbaum, Phil Karn, Neal Koblitz, Xuejia Lai, Tom Leranth, Mike Markowitz, Ralph Merkle, Bill Patton, Peter Pearson, Charles Pfleeger, Ken Pizzini, Bart Preneel, Mark Riordan, Joachim Schurman, and Marc Schwartz for reading and editing all or parts of the first edition; Marc Vauclair for translating the first edition into French; Abe Abraham, Ross Anderson, Dave Banisar, Steve Bellovin, Eli Biham, Matt Bishop, Matt Blaze, Gary Carter, Jan Camenisch, Claude Crépeau, Joan Daemen, Jorge Davila, Ed Dawson, Whit Diffie, Carl Ellison, Joan Feigenbaum, Niels Ferguson, Matt Franklin, Rosario Gennaro, Dieter Gollmann, Mark Goresky, Richard Graveman, Stuart Haber, Jingman He, Bob Hogue, Kenneth Iversen, Markus Jakobsson, Burt Kaliski, Phil Karn, John Kelsey, John Kennedy, Lars Knudsen, Paul Kocher, John Ladwig, Xuejia Lai, Arjen Lenstra, Paul Leyland, Mike Markowitz, Jim Massey, Bruce McNair, William Hugh Murray, Roger Needham, Clif Neuman, Kaisa Nyberg, Luke O'Connor, Peter Pearson, René Peralta, Bart Preneel, Yisrael Radai, Matt Robshaw, Michael Roe, Phil Rogaway, Avi Rubin, Paul Rubin, Selwyn Russell, Kazue Sako, Mahmoud Salmasizadeh, Markus Stadler, Dmitry Titov, Jimmy Upton, Marc Vauclair, Serge Vaudenay, Gideon Yuval, Glen Zorn, and several anonymous government employees for reading and editing all or parts of the second edition; Lawrie Brown, Leisa Condie, Joan Daemen, Peter Gutmann, Alan Insley, Chris Johnston, John Kelsey, Xuejia Lai, Bill Leininger, Mike Markowitz, Richard Outerbridge, Peter Pearson, Ken Pizzini, Colin Plumb, RSA Data Security, Inc., Michael Roe, Michael Wood, and Phil Zimmermann for providing source code; Paul MacNerland for creating the figures for the first edition; Karen Cooper for copyediting the second edition; Beth Friedman for proofreading the second edition; Carol Kennedy for indexing the second edition; the readers of sci.crypt and the Cypherpunks mailing list for commenting on ideas, answering questions, and finding errors in the first edition; Randy Seuss for providing Internet access; Jeff Duntemann and Jon Erickson for helping me get started; assorted random Insleys for the impetus, encouragement, support, conversations, friendship, and dinners; and AT&T Bell Labs for firing me and making this all possible. All these people helped to create a far better book than I could have created alone.

Bruce Schneier
Oak Park, Ill.
schneier@counterpane.com

About the Author

BRUCE SCHNEIER is an internationally renowned security technologist, called a "security guru" by *The Economist.* He is the author of twelve books — including his seminal work, *Applied Cryptography: Protocols, Algorithms, and Source Code in C,* and *Secrets & Lies: Digital Security in a Networked World* which has become a classic as well as hundreds of articles, essays, and academic papers. His influential newsletter "Crypto-Gram" and blog "Schneier on Security" are read by over 250,000 people. Schneier is a fellow at the Berkman Center for Internet and Society at Harvard Law School, a program fellow at the New America Foundation's Open Technology Institute, a board member of the Electronic Frontier Foundation, and an Advisory Board member of the Electronic Privacy Information Center. He is also the Chief Technology Officer of Resilient Systems, Inc. You can read his blog, essays, and academic papers at `www.schneier.com`. He tweets at `@schneierblog`.

CHAPTER 1

Foundations

1.1 TERMINOLOGY

Sender and Receiver

Suppose a sender wants to send a message to a receiver. Moreover, this sender wants to send the message securely: She wants to make sure an eavesdropper cannot read the message.

Messages and Encryption

A message is **plaintext** (sometimes called cleartext). The process of disguising a message in such a way as to hide its substance is **encryption.** An encrypted message is **ciphertext**. The process of turning ciphertext back into plaintext is **decryption**. This is all shown in Figure 1.1.

(If you want to follow the ISO 7498-2 standard, use the terms "encipher" and "decipher." It seems that some cultures find the terms "encrypt" and "decrypt" offensive, as they refer to dead bodies.)

The art and science of keeping messages secure is **cryptography**, and it is practiced by **cryptographers**. **Cryptanalysts** are practitioners of **cryptanalysis**, the art and science of breaking ciphertext; that is, seeing through the disguise. The branch of mathematics encompassing both cryptography and cryptanalysis is **cryptology** and its practitioners are **cryptologists**. Modern cryptologists are generally trained in theoretical mathematics—they have to be.

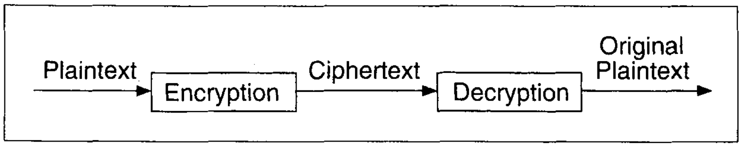


Figure 1.1 Encryption and Decryption.

Plaintext is denoted by M, for message, or P, for plaintext. It can be a stream of bits, a text file, a bitmap, a stream of digitized voice, a digital video image . . . whatever. As far as a computer is concerned, M is simply binary data. (After this chapter, this book concerns itself with binary data and computer cryptography.) The plaintext can be intended for either transmission or storage. In any case, M is the message to be encrypted.

Ciphertext is denoted by C. It is also binary data: sometimes the same size as M, sometimes larger. (By combining encryption with compression, C may be smaller than M. However, encryption does not accomplish this.) The encryption function E, operates on M to produce C. Or, in mathematical notation:

$$E(M) = C$$

In the reverse process, the decryption function D operates on C to produce M:

$$D(C) = M$$

Since the whole point of encrypting and then decrypting a message is to recover the original plaintext, the following identity must hold true:

$$D(E(M)) = M$$

Authentication, Integrity, and Nonrepudiation

In addition to providing confidentiality, cryptography is often asked to do other jobs:

- **Authentication**. It should be possible for the receiver of a message to ascertain its origin; an intruder should not be able to masquerade as someone else.
- **Integrity**. It should be possible for the receiver of a message to verify that it has not been modified in transit; an intruder should not be able to substitute a false message for a legitimate one.
- **Nonrepudiation**. A sender should not be able to falsely deny later that he sent a message.

These are vital requirements for social interaction on computers, and are analogous to face-to-face interactions. That someone is who he says he is . . . that someone's credentials—whether a driver's license, a medical degree, or a passport—are valid . . . that a document purporting to come from a person actually came from that person. . . . These are the things that authentication, integrity, and nonrepudiation provide.

Algorithms and Keys

A **cryptographic algorithm**, also called a **cipher**, is the mathematical function used for encryption and decryption. (Generally, there are two related functions: one for encryption and the other for decryption.)

If the security of an algorithm is based on keeping the way that algorithm works a secret, it is a **restricted** algorithm. Restricted algorithms have historical interest, but are woefully inadequate by today's standards. A large or changing group of users cannot use them, because every time a user leaves the group everyone else must switch to a different algorithm. If someone accidentally reveals the secret, everyone must change their algorithm.

Even more damning, restricted algorithms allow no quality control or standardization. Every group of users must have their own unique algorithm. Such a group can't use off-the-shelf hardware or software products; an eavesdropper can buy the same product and learn the algorithm. They have to write their own algorithms and implementations. If no one in the group is a good cryptographer, then they won't know if they have a secure algorithm.

Despite these major drawbacks, restricted algorithms are enormously popular for low-security applications. Users either don't realize or don't care about the security problems inherent in their system.

Modern cryptography solves this problem with a **key**, denoted by K. This key might be any one of a large number of values. The range of possible values of the key is called the **keyspace**. Both the encryption and decryption operations use this key (i.e., they are dependent on the key and this fact is denoted by the K subscript), so the functions now become:

$$E_K(M) = C$$
$$D_K(C) = M$$

Those functions have the property that (see Figure 1.2):

$$D_K(E_K(M)) = M$$

Some algorithms use a different encryption key and decryption key (see Figure 1.3). That is, the encryption key, K_1, is different from the corresponding decryption key, K_2. In this case:

$$E_{K_1}(M) = C$$
$$D_{K_2}(C) = M$$
$$D_{K_2}(E_{K_1}(M)) = M$$

All of the security in these algorithms is based in the key (or keys); none is based in the details of the algorithm. This means that the algorithm can be published and analyzed. Products using the algorithm can be mass-produced. It doesn't matter if an

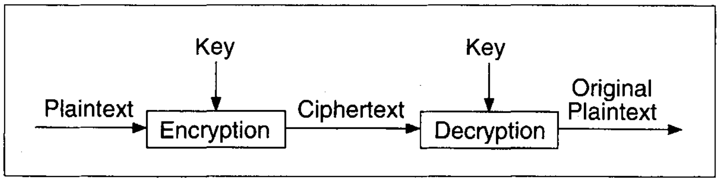


Figure 1.2 Encryption and decryption with a key.

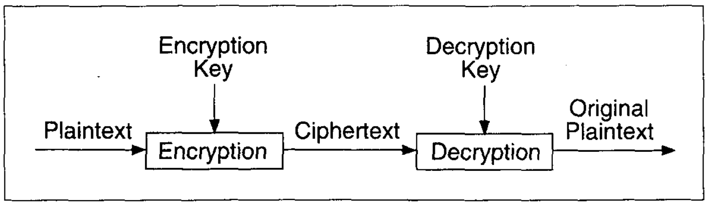


Figure 1.3 Encryption and decryption with two different keys.

eavesdropper knows your algorithm; if she doesn't know your particular key, she can't read your messages.

A **cryptosystem** is an algorithm, plus all possible plaintexts, ciphertexts, and keys.

Symmetric Algorithms

There are two general types of key-based algorithms: symmetric and public-key. **Symmetric algorithms**, sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa. In most symmetric algorithms, the encryption key and the decryption key are the same. These algorithms, also called secret-key algorithms, single-key algorithms, or one-key algorithms, require that the sender and receiver agree on a key before they can communicate securely. The security of a symmetric algorithm rests in the key; divulging the key means that anyone could encrypt and decrypt messages. As long as the communication needs to remain secret, the key must remain secret.

Encryption and decryption with a symmetric algorithm are denoted by:

$$E_K(M) = C$$
$$D_K(C) = M$$

Symmetric algorithms can be divided into two categories. Some operate on the plaintext a single bit (or sometimes byte) at a time; these are called **stream algorithms** or **stream ciphers**. Others operate on the plaintext in groups of bits. The groups of bits are called **blocks**, and the algorithms are called **block algorithms** or **block ciphers**. For modern computer algorithms, a typical block size is 64 bits—large enough to preclude analysis and small enough to be workable. (Before computers, algorithms generally operated on plaintext one character at a time. You can think of this as a stream algorithm operating on a stream of characters.)

Public-Key Algorithms

Public-key algorithms (also called asymmetric algorithms) are designed so that the key used for encryption is different from the key used for decryption. Furthermore, the decryption key cannot (at least in any reasonable amount of time) be calculated from the encryption key. The algorithms are called "public-key" because the encryption key can be made public: A complete stranger can use the encryption key to encrypt a message, but only a specific person with the corresponding decryption key can decrypt the message. In these systems, the encryption key is often called the **public key**, and the decryption key is often called the **private key**. The private key is sometimes also called the secret key, but to avoid confusion with symmetric algorithms, that tag won't be used here.

Encryption using public key K is denoted by:

$$E_K(M) = C$$

Even though the public key and private key are different, decryption with the corresponding private key is denoted by:

$$D_K(C) = M$$

Sometimes, messages will be encrypted with the private key and decrypted with the public key; this is used in digital signatures (see Section 2.6). Despite the possible confusion, these operations are denoted by, respectively:

$$E_K(M) = C$$
$$D_K(C) = M$$

Cryptanalysis

The whole point of cryptography is to keep the plaintext (or the key, or both) secret from eavesdroppers (also called adversaries, attackers, interceptors, interlopers, intruders, opponents, or simply the enemy). Eavesdroppers are assumed to have complete access to the communications between the sender and receiver.

Cryptanalysis is the science of recovering the plaintext of a message without access to the key. Successful cryptanalysis may recover the plaintext or the key. It also may find weaknesses in a cryptosystem that eventually lead to the previous results. (The loss of a key through noncryptanalytic means is called a **compromise**.)

An attempted cryptanalysis is called an **attack**. A fundamental assumption in cryptanalysis, first enunciated by the Dutchman A. Kerckhoffs in the nineteenth century, is that the secrecy must reside entirely in the key [794]. Kerckhoffs assumes that the cryptanalyst has complete details of the cryptographic algorithm and implementation. (Of course, one would assume that the CIA does not make a habit of telling Mossad about its cryptographic algorithms, but Mossad probably finds out anyway.) While real-world cryptanalysts don't always have such detailed information, it's a good assumption to make. If others can't break an algorithm, even with knowledge of how it works, then they certainly won't be able to break it without that knowledge.

There are four general types of cryptanalytic attacks. Of course, each of them assumes that the cryptanalyst has complete knowledge of the encryption algorithm used:

1. **Ciphertext-only attack**. The cryptanalyst has the ciphertext of several messages, all of which have been encrypted using the same encryption algorithm. The cryptanalyst's job is to recover the plaintext of as many messages as possible, or better yet to deduce the key (or keys) used to

encrypt the messages, in order to decrypt other messages encrypted with the same keys.

Given: $C_1 = E_k(P_1), C_2 = E_k(P_2), \ldots C_i = E_k(P_i)$
Deduce: Either $P_1, P_2, \ldots P_i$; k; or an algorithm
to infer P_{i+1} from $C_{i+1} = E_k(P_{i+1})$

2. **Known-plaintext attack**. The cryptanalyst has access not only to the ciphertext of several messages, but also to the plaintext of those messages. His job is to deduce the key (or keys) used to encrypt the messages or an algorithm to decrypt any new messages encrypted with the same key (or keys).

   Given: $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2), \ldots P_i, C_i = E_k(P_i)$
   Deduce: Either k, or an algorithm
   to infer P_{i+1} from $C_{i+1} = E_k(P_{i+1})$

3. **Chosen-plaintext attack**. The cryptanalyst not only has access to the ciphertext and associated plaintext for several messages, but he also chooses the plaintext that gets encrypted. This is more powerful than a known-plaintext attack, because the cryptanalyst can choose specific plaintext blocks to encrypt, ones that might yield more information about the key. His job is to deduce the key (or keys) used to encrypt the messages or an algorithm to decrypt any new messages encrypted with the same key (or keys).

   Given: $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2), \ldots P_i, C_i = E_k(P_i)$,
   where the cryptanalyst gets to choose $P_1, P_2, \ldots P_i$
   Deduce: Either k, or an algorithm to infer P_{i+1} from $C_{i+1} = E_k(P_{i+1})$

4. **Adaptive-chosen-plaintext attack**. This is a special case of a chosen-plaintext attack. Not only can the cryptanalyst choose the plaintext that is encrypted, but he can also modify his choice based on the results of previous encryption. In a chosen-plaintext attack, a cryptanalyst might just be able to choose one large block of plaintext to be encrypted; in an adaptive-chosen-plaintext attack he can choose a smaller block of plaintext and then choose another based on the results of the first, and so forth.

There are at least three other types of cryptanalytic attack.

5. **Chosen-ciphertext attack**. The cryptanalyst can choose different ciphertexts to be decrypted and has access to the decrypted plaintext. For example, the cryptanalyst has access to a tamperproof box that does automatic decryption. His job is to deduce the key.

   Given: $C_1, P_1 = D_k(C_1), C_2, P_2 = D_k(C_2), \ldots C_i, P_i = D_k(C_i)$
   Deduce: k

This attack is primarily applicable to public-key algorithms and will be discussed in Section 19.3. A chosen-ciphertext attack is sometimes effective against a symmetric algorithm as well. (Sometimes a chosen-plaintext attack and a chosen-ciphertext attack are together known as a **chosen-text attack**.)

6. **Chosen-key attack**. This attack doesn't mean that the cryptanalyst can choose the key; it means that he has some knowledge about the relationship between different keys. It's strange and obscure, not very practical, and discussed in Section 12.4.
7. **Rubber-hose cryptanalysis**. The cryptanalyst threatens, blackmails, or tortures someone until they give him the key. Bribery is sometimes referred to as a **purchase-key attack**. These are all very powerful attacks and often the best way to break an algorithm.

Known-plaintext attacks and chosen-plaintext attacks are more common than you might think. It is not unheard-of for a cryptanalyst to get a plaintext message that has been encrypted or to bribe someone to encrypt a chosen message. You may not even have to bribe someone; if you give a message to an ambassador, you will probably find that it gets encrypted and sent back to his country for consideration. Many messages have standard beginnings and endings that might be known to the cryptanalyst. Encrypted source code is especially vulnerable because of the regular appearance of keywords: #define, struct, else, return. Encrypted executable code has the same kinds of problems: functions, loop structures, and so on. Known-plaintext attacks (and even chosen-plaintext attacks) were successfully used against both the Germans and the Japanese during World War II. David Kahn's books [794,795,796] have historical examples of these kinds of attacks.

And don't forget Kerckhoffs's assumption: If the strength of your new cryptosystem relies on the fact that the attacker does not know the algorithm's inner workings, you're sunk. If you believe that keeping the algorithm's insides secret improves the security of your cryptosystem more than letting the academic community analyze it, you're wrong. And if you think that someone won't disassemble your code and reverse-engineer your algorithm, you're naïve. (In 1994 this happened with the RC4 algorithm—see Section 17.1.) The best algorithms we have are the ones that have been made public, have been attacked by the world's best cryptographers for years, and are still unbreakable. (The National Security Agency keeps their algorithms secret from outsiders, but they have the best cryptographers in the world working within their walls—you don't. Additionally, they discuss their algorithms with one another, relying on peer review to uncover any weaknesses in their work.)

Cryptanalysts don't always have access to the algorithms, as when the United States broke the Japanese diplomatic code PURPLE during World War II [794]—but they often do. If the algorithm is being used in a commercial security program, it is simply a matter of time and money to disassemble the program and recover the algorithm. If the algorithm is being used in a military communications system, it is simply a matter of time and money to buy (or steal) the equipment and reverse-engineer the algorithm.

Those who claim to have an unbreakable cipher simply because they can't break it are either geniuses or fools. Unfortunately, there are more of the latter in the world. Beware of people who extol the virtues of their algorithms, but refuse to make them public; trusting their algorithms is like trusting snake oil.

Good cryptographers rely on peer review to separate the good algorithms from the bad.

Security of Algorithms

Different algorithms offer different degrees of security; it depends on how hard they are to break. If the cost required to break an algorithm is greater than the value of the encrypted data, then you're probably safe. If the time required to break an algorithm is longer than the time the encrypted data must remain secret, then you're probably safe. If the amount of data encrypted with a single key is less than the amount of data necessary to break the algorithm, then you're probably safe.

I say "probably" because there is always a chance of new breakthroughs in cryptanalysis. On the other hand, the value of most data decreases over time. It is important that the value of the data always remain less than the cost to break the security protecting it.

Lars Knudsen classified these different categories of breaking an algorithm. In decreasing order of severity [858]:

1. **Total break.** A cryptanalyst finds the key, K, such that $D_K(C) = P$.
2. **Global deduction.** A cryptanalyst finds an alternate algorithm, A, equivalent to $D_K(C)$, without knowing K.
3. **Instance (or local) deduction.** A cryptanalyst finds the plaintext of an intercepted ciphertext.
4. **Information deduction.** A cryptanalyst gains some information about the key or plaintext. This information could be a few bits of the key, some information about the form of the plaintext, and so forth.

An algorithm is **unconditionally secure** if, no matter how much ciphertext a cryptanalyst has, there is not enough information to recover the plaintext. In point of fact, only a one-time pad (see Section 1.5) is unbreakable given infinite resources. All other cryptosystems are breakable in a ciphertext-only attack, simply by trying every possible key one by one and checking whether the resulting plaintext is meaningful. This is called a **brute-force** attack (see Section 7.1).

Cryptography is more concerned with cryptosystems that are computationally infeasible to break. An algorithm is considered **computationally secure** (sometimes called strong) if it cannot be broken with available resources, either current or future. Exactly what constitutes "available resources" is open to interpretation.

You can measure the complexity (see Section 11.1) of an attack in different ways:

1. **Data complexity**. The amount of data needed as input to the attack.
2. **Processing complexity**. The time needed to perform the attack. This is often called the **work factor**.
3. **Storage requirements**. The amount of memory needed to do the attack.

As a rule of thumb, the complexity of an attack is taken to be the minimum of these three factors. Some attacks involve trading off the three complexities: A faster attack might be possible at the expense of a greater storage requirement.

Complexities are expressed as orders of magnitude. If an algorithm has a processing complexity of 2^{128}, then 2^{128} operations are required to break the algorithm. (These operations may be complex and time-consuming.) Still, if you assume that you have enough computing speed to perform a million operations every second and you set a million parallel processors against the task, it will still take over 10^{19} years to recover the key. That's a billion times the age of the universe.

While the complexity of an attack is constant (until some cryptanalyst finds a better attack, of course), computing power is anything but. There have been phenomenal advances in computing power during the last half-century and there is no reason to think this trend won't continue. Many cryptanalytic attacks are perfect for parallel machines: The task can be broken down into billions of tiny pieces and none of the processors need to interact with each other. Pronouncing an algorithm secure simply because it is infeasible to break, given current technology, is dicey at best. Good cryptosystems are designed to be infeasible to break with the computing power that is expected to evolve many years in the future.

Historical Terms

Historically, a **code** refers to a cryptosystem that deals with linguistic units: words, phrases, sentences, and so forth. For example, the word "OCELOT" might be the ciphertext for the entire phrase "TURN LEFT 90 DEGREES," the word "LOLLIPOP" might be the ciphertext for "TURN RIGHT 90 DEGREES," and the words "BENT EAR" might be the ciphertext for "HOWITZER." Codes of this type are not discussed in this book; see [794,795]. Codes are only useful for specialized circumstances. Ciphers are useful for any circumstance. If your code has no entry for "ANTEATERS," then you can't say it. You can say anything with a cipher.

1.2 STEGANOGRAPHY

Steganography serves to hide secret messages in other messages, such that the secret's very existence is concealed. Generally the sender writes an innocuous message and then conceals a secret message on the same piece of paper. Historical tricks include invisible inks, tiny pin punctures on selected characters, minute differences between handwritten characters, pencil marks on typewritten characters, grilles which cover most of the message except for a few characters, and so on.

More recently, people are hiding secret messages in graphic images. Replace the least significant bit of each byte of the image with the bits of the message. The graphical image won't change appreciably—most graphics standards specify more gradations of color than the human eye can notice—and the message can be stripped out at the receiving end. You can store a 64-kilobyte message in a 1024 × 1024 grey-scale picture this way. Several public-domain programs do this sort of thing.

Peter Wayner's **mimic functions** obfuscate messages. These functions modify a message so that its statistical profile resembles that of something else: the classifieds section of *The New York Times*, a play by Shakespeare, or a newsgroup on the Internet [1584,1585]. This type of steganography won't fool a person, but it might fool some big computers scanning the Internet for interesting messages.

1.3 SUBSTITUTION CIPHERS AND TRANSPOSITION CIPHERS

Before computers, cryptography consisted of character-based algorithms. Different cryptographic algorithms either substituted characters for one another or transposed characters with one another. The better algorithms did both, many times each.

Things are more complex these days, but the philosophy remains the same. The primary change is that algorithms work on bits instead of characters. This is actually just a change in the alphabet size: from 26 elements to two elements. Most good cryptographic algorithms still combine elements of substitution and transposition.

Substitution Ciphers

A **substitution cipher** is one in which each character in the plaintext is substituted for another character in the ciphertext. The receiver inverts the substitution on the ciphertext to recover the plaintext.

In classical cryptography, there are four types of substitution ciphers:

- A **simple substitution cipher**, or **monoalphabetic cipher**, is one in which each character of the plaintext is replaced with a corresponding character of ciphertext. The cryptograms in newspapers are simple substitution ciphers.
- A **homophonic substitution cipher** is like a simple substitution cryptosystem, except a single character of plaintext can map to one of several characters of ciphertext. For example, "A" could correspond to either 5, 13, 25, or 56, "B" could correspond to either 7, 19, 31, or 42, and so on.
- A **polygram substitution cipher** is one in which blocks of characters are encrypted in groups. For example, "ABA" could correspond to "RTQ," "ABB" could correspond to "SLL," and so on.
- A **polyalphabetic substitution cipher** is made up of multiple simple substitution ciphers. For example, there might be five different simple substitution ciphers used; the particular one used changes with the position of each character of the plaintext.

The famous **Caesar Cipher**, in which each plaintext character is replaced by the character three to the right modulo 26 ("A" is replaced by "D," "B" is replaced by "E," . . . , "W" is replaced by "Z," "X" is replaced by "A," "Y" is replaced by "B," and "Z" is replaced by "C") is a simple substitution cipher. It's actually even simpler, because the ciphertext alphabet is a rotation of the plaintext alphabet and not an arbitrary permutation.

ROT13 is a simple encryption program commonly found on UNIX systems; it is also a simple substitution cipher. In this cipher, "A" is replaced by "N," "B" is replaced by "O," and so on. Every letter is rotated 13 places.

Encrypting a file twice with ROT13 restores the original file.

$$P = \mathrm{ROT13}\ (\mathrm{ROT13}\ (P))$$

ROT13 is not intended for security; it is often used in Usenet posts to hide potentially offensive text, to avoid giving away the solution to a puzzle, and so forth.

Simple substitution ciphers can be easily broken because the cipher does not hide the underlying frequencies of the different letters of the plaintext. All it takes is about 25 English characters before a good cryptanalyst can reconstruct the plaintext [1434]. An algorithm for solving these sorts of ciphers can be found in [578,587, 1600,78,1475,1236,880]. A good computer algorithm is [703].

Homophonic substitution ciphers were used as early as 1401 by the Duchy of Mantua [794]. They are much more complicated to break than simple substitution ciphers, but still do not obscure all of the statistical properties of the plaintext language. With a known-plaintext attack, the ciphers are trivial to break. A ciphertext-only attack is harder, but only takes a few seconds on a computer. Details are in [1261].

Polygram substitution ciphers are ciphers in which groups of letters are encrypted together. The Playfair cipher, invented in 1854, was used by the British during World War I [794]. It encrypts pairs of letters together. Its cryptanalysis is discussed in [587,1475,880]. The Hill cipher is another example of a polygram substitution cipher [732]. Sometimes you see Huffman coding used as a cipher; this is an insecure polygram substitution cipher.

Polyalphabetic substitution ciphers were invented by Leon Battista in 1568 [794]. They were used by the Union army during the American Civil War. Despite the fact that they can be broken easily [819,577,587,794] (especially with the help of computers), many commercial computer security products use ciphers of this form [1387,1390,1502]. (Details on how to break this encryption scheme, as used in WordPerfect, can be found in [135,139].) The Vigenère cipher, first published in 1586, and the Beaufort cipher are also examples of polyalphabetic substitution ciphers.

Polyalphabetic substitution ciphers have multiple one-letter keys, each of which is used to encrypt one letter of the plaintext. The first key encrypts the first letter of the plaintext, the second key encrypts the second letter of the plaintext, and so on. After all the keys are used, the keys are recycled. If there were 20 one-letter keys, then every twentieth letter would be encrypted with the same key. This is called the **period** of the cipher. In classical cryptography, ciphers with longer periods were significantly harder to break than ciphers with short periods. There are computer techniques that can easily break substitution ciphers with very long periods.

A **running-key cipher**—sometimes called a book cipher—in which one text is used to encrypt another text, is another example of this sort of cipher. Even though this cipher has a period the length of the text, it can also be broken easily [576,794].

Transposition Ciphers

In a **transposition cipher** the plaintext remains the same, but the order of characters is shuffled around. In a **simple columnar transposition cipher**, the plaintext is written horizontally onto a piece of graph paper of fixed width and the ciphertext is read off vertically (see Figure 1.4). Decryption is a matter of writing the ciphertext vertically onto a piece of graph paper of identical width and then reading the plaintext off horizontally.

Cryptanalysis of these ciphers is discussed in [587,1475]. Since the letters of the ciphertext are the same as those of the plaintext, a frequency analysis on the ciphertext would reveal that each letter has approximately the same likelihood as in English. This gives a very good clue to a cryptanalyst, who can then use a variety of techniques to determine the right ordering of the letters to obtain the plaintext. Putting the ciphertext through a second transposition cipher greatly enhances security. There are even more complicated transposition ciphers, but computers can break almost all of them.

The German ADFGVX cipher, used during World War I, is a transposition cipher combined with a simple substitution. It was a very complex algorithm for its day but was broken by Georges Painvin, a French cryptanalyst [794].

Although many modern algorithms use transposition, it is troublesome because it requires a lot of memory and sometimes requires messages to be only certain lengths. Substitution is far more common.

Rotor Machines

In the 1920s, various mechanical encryption devices were invented to automate the process of encryption. Most were based on the concept of a **rotor**, a mechanical wheel wired to perform a general substitution.

A **rotor machine** has a keyboard and a series of rotors, and implements a version of the Vigenère cipher. Each rotor is an arbitrary permutation of the alphabet, has 26 positions, and performs a simple substitution. For example, a rotor might be wired

Plaintext: COMPUTER GRAPHICS MAY BE SLOW BUT AT LEAST IT'S EXPENSIVE.

```
COMPUTERGR
APHICSMAYB
ESLOWBUTAT
LEASTITSEX
PENSIVE
```

Ciphertext: CAELP OPSEE MHLAN PIOSS UCWTI TSBIV EMUTE RATSG YAERB TX

Figure 1.4 Columnar transposition cipher.

to substitute "F" for "A," "U" for "B," "L" for "C," and so on. And the output pins of one rotor are connected to the input pins of the next.

For example, in a 4-rotor machine the first rotor might substitute "F" for "A," the second might substitute "Y" for "F," the third might substitute "E" for "Y," and the fourth might substitute "C" for "E"; "C" would be the output ciphertext. Then some of the rotors shift, so next time the substitutions will be different.

It is the combination of several rotors and the gears moving them that makes the machine secure. Because the rotors all move at different rates, the period for an n-rotor machine is 26^n. Some rotor machines can also have a different number of positions on each rotor, further frustrating cryptanalysis.

The best-known rotor device is the Enigma. The Enigma was used by the Germans during World War II. The idea was invented by Arthur Scherbius and Arvid Gerhard Damm in Europe. It was patented in the United States by Arthur Scherbius [1383]. The Germans beefed up the basic design considerably for wartime use.

The German Enigma had three rotors, chosen from a set of five, a plugboard that slightly permuted the plaintext, and a reflecting rotor that caused each rotor to operate on each plaintext letter twice. As complicated as the Enigma was, it was broken during World War II. First, a team of Polish cryptographers broke the German Enigma and explained their attack to the British. The Germans modified their Enigma as the war progressed, and the British continued to cryptanalyze the new versions. For explanations of how rotor ciphers work and how they were broken, see [794,86,448,498,446,880,1315,1587,690]. Two fascinating accounts of how the Enigma was broken are [735,796].

Further Reading

This is not a book about classical cryptography, so I will not dwell further on these subjects. Two excellent precomputer cryptology books are [587,1475]; [448] presents some modern cryptanalysis of cipher machines. Dorothy Denning discusses many of these ciphers in [456] and [880] has some fairly complex mathematical analysis of the same ciphers. Another older cryptography text, which discusses analog cryptography, is [99]. An article that presents a good overview of the subject is [579]. David Kahn's historical cryptography books are also excellent [794,795,796].

1.4 SIMPLE XOR

XOR is exclusive-or operation: '^' in C or $\oplus$ in mathematical notation. It's a standard operation on bits:

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

Also note that:

$$a \oplus a = 0$$
$$a \oplus b \oplus b = a$$

The simple-XOR algorithm is really an embarrassment; it's nothing more than a Vigenère polyalphabetic cipher. It's here only because of its prevalence in commercial software packages, at least those in the MS-DOS and Macintosh worlds [1502,1387]. Unfortunately, if a software security program proclaims that it has a "proprietary" encryption algorithm—significantly faster than DES—the odds are that it is some variant of this.

```
/* Usage:  crypto key input_file output_file */

void main (int argc, char *argv[])
{
    FILE *fi, *fo;
    char *cp;
    int c;

    if ((cp = argv[1]) && *cp!='\0')  {
        if ((fi = fopen(argv[2], "rb")) != NULL)  {
            if ((fo = fopen(argv[3], "wb")) != NULL)  {
                while ((c = getc(fi)) != EOF)  {
                    if (!*cp) cp = argv[1];
                    c ^= *(cp++);
                    putc(c,fo);
                }
                fclose(fo);
            }
            fclose(fi);
        }
    }
}
```

This is a symmetric algorithm. The plaintext is being XORed with a keyword to generate the ciphertext. Since XORing the same value twice restores the original, encryption and decryption use exactly the same program:

$$P \oplus K = C$$

$$C \oplus K = P$$

There's no real security here. This kind of encryption is trivial to break, even without computers [587,1475]. It will only take a few seconds with a computer.

Assume the plaintext is English. Furthermore, assume the key length is any small number of bytes. Here's how to break it:

1. Discover the length of the key by a procedure known as **counting coincidences** [577]. XOR the ciphertext against itself shifted various numbers of bytes, and count those bytes that are equal. If the displacement is a multiple of the key length, then something over 6 percent of the bytes will be equal. If it is not, then less than 0.4 percent will be equal (assuming a random key encrypting normal ASCII text; other plaintext will have different numbers). This is called the **index of coincidence**. The smallest displacement that indicates a multiple of the key length is the length of the key.

2. Shift the ciphertext by that length and XOR it with itself. This removes the key and leaves you with plaintext XORed with the plaintext shifted the length of the key. Since English has 1.3 bits of real information per byte (see Section 11.1), there is plenty of redundancy for determining a unique decryption.

Despite this, the list of software vendors that tout this toy algorithm as being "almost as secure as DES" is staggering [1387]. It is the algorithm (with a 160-bit repeated "key") that the NSA finally allowed the U.S. digital cellular phone industry to use for voice privacy. An XOR might keep your kid sister from reading your files, but it won't stop a cryptanalyst for more than a few minutes.

1.5 ONE-TIME PADS

Believe it or not, there is a perfect encryption scheme. It's called a **one-time pad**, and was invented in 1917 by Major Joseph Mauborgne and AT&T's Gilbert Vernam [794]. (Actually, a one-time pad is a special case of a threshold scheme; see Section 3.7.) Classically, a one-time pad is nothing more than a large nonrepeating set of truly random key letters, written on sheets of paper, and glued together in a pad. In its original form, it was a one-time tape for teletypewriters. The sender uses each key letter on the pad to encrypt exactly one plaintext character. Encryption is the addition modulo 26 of the plaintext character and the one-time pad key character.

Each key letter is used exactly once, for only one message. The sender encrypts the message and then destroys the used pages of the pad or used section of the tape. The receiver has an identical pad and uses each key on the pad, in turn, to decrypt each letter of the ciphertext. The receiver destroys the same pad pages or tape section after decrypting the message. New message—new key letters. For example, if the message is:

```
ONETIMEPAD
```

and the key sequence from the pad is

```
TBFRGFARFM
```

then the ciphertext is

```
IPKLPSFHGQ
```

because

$$O + T \bmod 26 = I$$
$$N + B \bmod 26 = P$$
$$E + F \bmod 26 = K$$

etc.

Assuming an eavesdropper can't get access to the one-time pad used to encrypt the message, this scheme is perfectly secure. A given ciphertext message is equally likely to correspond to any possible plaintext message of equal size.

Since every key sequence is equally likely (remember, the key letters are generated randomly), an adversary has no information with which to cryptanalyze the ciphertext. The key sequence could just as likely be:

```
POYYAEAAZX
```

which would decrypt to:

```
SALMONEGGS
```

or

```
BXFGBMTMXM
```

which would decrypt to:

```
GREENFLUID
```

This point bears repeating: Since every plaintext message is equally possible, there is no way for the cryptanalyst to determine which plaintext message is the correct one. A random key sequence added to a nonrandom plaintext message produces a completely random ciphertext message and no amount of computing power can change that.

The caveat, and this is a big one, is that the key letters have to be generated randomly. Any attacks against this scheme will be against the method used to generate the key letters. Using a pseudo-random number generator doesn't count; they always have nonrandom properties. If you use a real random source—this is much harder than it might first appear, see Section 17.14—it's secure.

The other important point is that you can never use the key sequence again, ever. Even if you use a multiple-gigabyte pad, if a cryptanalyst has multiple ciphertexts whose keys overlap, he can reconstruct the plaintext. He slides each pair of ciphertexts against each other and counts the number of matches at each position. If they are aligned right, the proportion of matches jumps suddenly—the exact percentages depend on the plaintext language. From this point cryptanalysis is easy. It's like the index of coincidence, but with just two "periods" to compare [904]. Don't do it.

The idea of a one-time pad can be easily extended to binary data. Instead of a one-time pad consisting of letters, use a one-time pad of bits. Instead of adding the plaintext to the one-time pad, use an XOR. To decrypt, XOR the ciphertext with the same one-time pad. Everything else remains the same and the security is just as perfect.

This all sounds good, but there are a few problems. Since the key bits must be random and can never be used again, the length of the key sequence must be equal to the length of the message. A one-time pad might be suitable for a few short messages, but it will never work for a 1.544 Mbps communications channel. You can store 650 megabytes worth of random bits on a CD-ROM, but there are problems. First, you want exactly two copies of the random bits, but CD-ROMs are economical only for large quantities. And second, you want to be able to destroy the bits already used. CD-ROM has no erase facilities except for physically destroying the entire disk. Digital tape is a much better medium for this sort of thing.

Even if you solve the key distribution and storage problem, you have to make sure the sender and receiver are perfectly synchronized. If the receiver is off by a bit (or if some bits are dropped during the transmission), the message won't make any sense. On the other hand, if some bits are altered during transmission (without any bits being added or removed—something far more likely to happen due to random noise), only those bits will be decrypted incorrectly. But on the other hand, a one-time pad provides no authenticity.

One-time pads have applications in today's world, primarily for ultra-secure low-bandwidth channels. The hotline between the United States and the former Soviet Union was (is it still active?) rumored to be encrypted with a one-time pad. Many Soviet spy messages to agents were encrypted using one-time pads. These messages are still secure today and will remain that way forever. It doesn't matter how long the supercomputers work on the problem. Even after the aliens from Andromeda land with their massive spaceships and undreamed-of computing power, they will not be able to read the Soviet spy messages encrypted with one-time pads (unless they can also go back in time and get the one-time pads).

1.6 COMPUTER ALGORITHMS

There are many cryptographic algorithms. These are three of the most common:

- DES (Data Encryption Standard) is the most popular computer encryption algorithm. DES is a U.S. and international standard. It is a symmetric algorithm; the same key is used for encryption and decryption.
- RSA (named for its creators—Rivest, Shamir, and Adleman) is the most popular public-key algorithm. It can be used for both encryption and digital signatures.
- DSA (Digital Signature Algorithm, used as part of the Digital Signature Standard) is another public-key algorithm. It cannot be used for encryption, but only for digital signatures.

These are the kinds of stuff this book is about.

1.7 LARGE NUMBERS

Throughout this book, I use various large numbers to describe different things in cryptography. Because it is so easy to lose sight of these numbers and what they signify, Table 1.1 gives physical analogues for some of them.

These numbers are order-of-magnitude estimates, and have been culled from a variety of sources. Many of the astrophysics numbers are explained in Freeman

TABLE 1.1
Large Numbers

Physical Analogue	Number
Odds of being killed by lightning (per day)	1 in 9 billion (2^{33})
Odds of winning the top prize in a U.S. state lottery	1 in 4,000,000 (2^{22})
Odds of winning the top prize in a U.S. state lottery and being killed by lightning in the same day	1 in 2^{55}
Odds of drowning (in the U.S. per year)	1 in 59,000 (2^{16})
Odds of being killed in an automobile accident (in the U.S. in 1993)	1 in 6100 (2^{13})
Odds of being killed in an automobile accident (in the U.S. per lifetime)	1 in 88 (2^{7})
Time until the next ice age	14,000 (2^{14}) years
Time until the sun goes nova	10^{9} (2^{30}) years
Age of the planet	10^{9} (2^{30}) years
Age of the Universe	10^{10} (2^{34}) years
Number of atoms in the planet	10^{51} (2^{170})
Number of atoms in the sun	10^{57} (2^{190})
Number of atoms in the galaxy	10^{67} (2^{223})
Number of atoms in the Universe (dark matter excluded)	10^{77} (2^{265})
Volume of the Universe	10^{84} (2^{280}) cm^{3}
If the Universe is Closed:	
Total lifetime of the Universe	10^{11} (2^{37}) years 10^{18} (2^{61}) seconds
If the Universe is Open:	
Time until low-mass stars cool off	10^{14} (2^{47}) years
Time until planets detach from stars	10^{15} (2^{50}) years
Time until stars detach from galaxies	10^{19} (2^{64}) years
Time until orbits decay by gravitational radiation	10^{20} (2^{67}) years
Time until black holes decay by the Hawking process	10^{64} (2^{213}) years
Time until all matter is liquid at zero temperature	10^{65} (2^{216}) years
Time until all matter decays to iron	$10^{10^{26}}$ years
Time until all matter collapses to black holes	$10^{10^{76}}$ years

Dyson's paper, "Time Without End: Physics and Biology in an Open Universe," in *Reviews of Modern Physics*, v. 52, n. 3, July 1979, pp. 447–460. Automobile accident deaths are calculated from the Department of Transportation's statistic of 163 deaths per million people in 1993 and an average lifespan of 69.7 years.

PART I

CRYPTOGRAPHIC PROTOCOLS

CHAPTER 2

Protocol Building Blocks

2.1 INTRODUCTION TO PROTOCOLS

The whole point of cryptography is to solve problems. (Actually, that's the whole point of computers—something many people tend to forget.) Cryptography solves problems that involve secrecy, authentication, integrity, and dishonest people. You can learn all about cryptographic algorithms and techniques, but these are academic unless they can solve a problem. This is why we are going to look at protocols first.

A **protocol** is a series of steps, involving two or more parties, designed to accomplish a task. This is an important definition. A "series of steps" means that the protocol has a sequence, from start to finish. Every step must be executed in turn, and no step can be taken before the previous step is finished. "Involving two or more parties" means that at least two people are required to complete the protocol; one person alone does not make a protocol. A person alone can perform a series of steps to accomplish a task (like baking a cake), but this is not a protocol. (Someone else must eat the cake to make it a protocol.) Finally, "designed to accomplish a task" means that the protocol must achieve something. Something that looks like a protocol but does not accomplish a task is not a protocol—it's a waste of time.

Protocols have other characteristics as well:

- Everyone involved in the protocol must know the protocol and all of the steps to follow in advance.
- Everyone involved in the protocol must agree to follow it.
- The protocol must be unambiguous; each step must be well defined and there must be no chance of a misunderstanding.
- The protocol must be complete; there must be a specified action for every possible situation.

The protocols in this book are organized as a series of steps. Execution of the protocol proceeds linearly through the steps, unless there are instructions to branch to another step. Each step involves at least one of two things: computations by one or more of the parties, or messages sent among the parties.

A **cryptographic protocol** is a protocol that uses cryptography. The parties can be friends and trust each other implicitly or they can be adversaries and not trust one another to give the correct time of day. A cryptographic protocol involves some cryptographic algorithm, but generally the goal of the protocol is something beyond simple secrecy. The parties participating in the protocol might want to share parts of their secrets to compute a value, jointly generate a random sequence, convince one another of their identity, or simultaneously sign a contract. The whole point of using cryptography in a protocol is to prevent or detect eavesdropping and cheating. If you have never seen these protocols before, they will radically change your ideas of what mutually distrustful parties can accomplish over a computer network. In general, this can be stated as:

— It should not be possible to do more or learn more than what is specified in the protocol.

This is a lot harder than it looks. In the next few chapters I discuss a lot of protocols. In some of them it is possible for one of the participants to cheat the other. In others, it is possible for an eavesdropper to subvert the protocol or learn secret information. Some protocols fail because the designers weren't thorough enough in their requirements definitions. Others fail because their designers weren't thorough enough in their analysis. Like algorithms, it is much easier to prove insecurity than it is to prove security.

The Purpose of Protocols

In daily life, there are informal protocols for almost everything: ordering goods over the telephone, playing poker, voting in an election. No one thinks much about these protocols; they have evolved over time, everyone knows how to use them, and they work reasonably well.

These days, more and more human interaction takes place over computer networks instead of face-to-face. Computers need formal protocols to do the same things that people do without thinking. If you moved from one state to another and found a voting booth that looked completely different from the ones you were used to, you could easily adapt. Computers are not nearly so flexible.

Many face-to-face protocols rely on people's presence to ensure fairness and security. Would you send a stranger a pile of cash to buy groceries for you? Would you play poker with someone if you couldn't see him shuffle and deal? Would you mail the government your secret ballot without some assurance of anonymity?

It is naïve to assume that people on computer networks are honest. It is naïve to assume that the managers of computer networks are honest. It is even naïve to assume that the designers of computer networks are honest. Most are, but the dishonest few can do a lot of damage. By formalizing protocols, we can examine ways in which dishonest parties can subvert them. Then we can develop protocols that are immune to that subversion.

In addition to formalizing behavior, protocols abstract the process of accomplishing a task from the mechanism by which the task is accomplished. A communications protocol is the same whether implemented on PCs or VAXs. We can examine the protocol without getting bogged down in the implementation details. When we are convinced we have a good protocol, we can implement it in everything from computers to telephones to intelligent muffin toasters.

The Players

To help demonstrate protocols, I have enlisted the aid of several people (see Table 2.1). Alice and Bob are the first two. They will perform all general two-person protocols. As a rule, Alice will initiate all protocols and Bob will respond. If the protocol requires a third or fourth person, Carol and Dave will perform those roles. Other actors will play specialized supporting roles; they will be introduced later.

Arbitrated Protocols

An **arbitrator** is a disinterested third party trusted to complete a protocol (see Figure 2.1a). Disinterested means that the arbitrator has no vested interest in the protocol and no particular allegiance to any of the parties involved. Trusted means that all people involved in the protocol accept what he says as true, what he does as correct, and that he will complete his part of the protocol. Arbitrators can help complete protocols between two mutually distrustful parties.

In the real world, lawyers are often used as arbitrators. For example, Alice is selling a car to Bob, a stranger. Bob wants to pay by check, but Alice has no way of knowing if the check is good. Alice wants the check to clear before she turns the title over to Bob. Bob, who doesn't trust Alice any more than she trusts him, doesn't want to hand over a check without receiving a title.

TABLE 2.1
Dramatis Personae

Alice	First participant in all the protocols
Bob	Second participant in all the protocols
Carol	Participant in the three- and four-party protocols
Dave	Participant in the four-party protocols
Eve	Eavesdropper
Mallory	Malicious active attacker
Trent	Trusted arbitrator
Walter	Warden; he'll be guarding Alice and Bob in some protocols
Peggy	Prover
Victor	Verifier

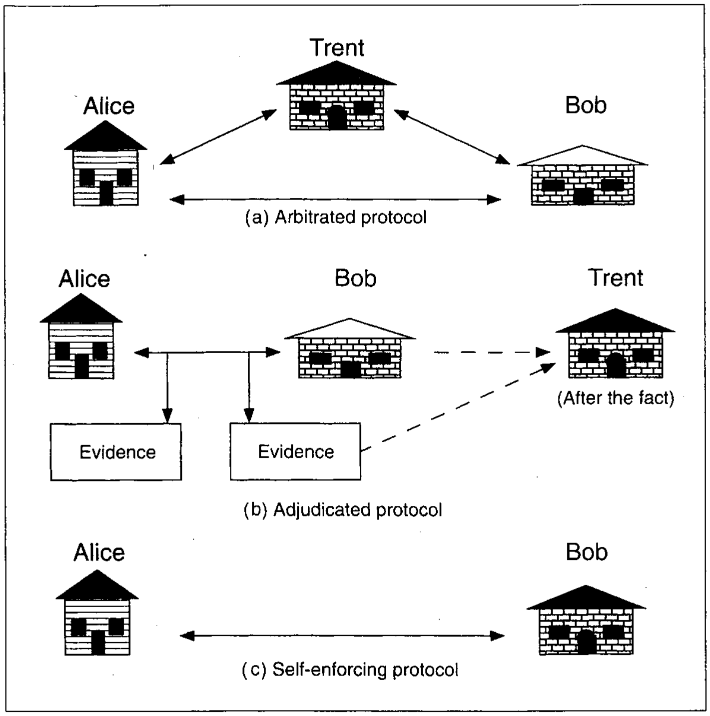


Figure 2.1 Types of protocols.

Enter a lawyer trusted by both. With his help, Alice and Bob can use the following protocol to ensure that neither cheats the other:

(1) Alice gives the title to the lawyer.
(2) Bob gives the check to Alice.
(3) Alice deposits the check.
(4) After waiting a specified time period for the check to clear, the lawyer gives the title to Bob. If the check does not clear within the specified time period, Alice shows proof of this to the lawyer and the lawyer returns the title to Alice.

In this protocol, Alice trusts the lawyer not to give Bob the title unless the check has cleared, and to give it back to her if the check does not clear. Bob trusts the lawyer to hold the title until the check clears, and to give it to him once it does. The lawyer doesn't care if the check clears. He will do his part of the protocol in either case, because he will be paid in either case.

In the example, the lawyer is playing the part of an escrow agent. Lawyers also act as arbitrators for wills and sometimes for contract negotiations. The various stock exchanges act as arbitrators between buyers and sellers.

Bankers also arbitrate protocols. Bob can use a certified check to buy a car from Alice:

(1) Bob writes a check and gives it to the bank.
(2) After putting enough of Bob's money on hold to cover the check, the bank certifies the check and gives it back to Bob.
(3) Alice gives the title to Bob and Bob gives the certified check to Alice.
(4) Alice deposits the check.

This protocol works because Alice trusts the banker's certification. Alice trusts the bank to hold Bob's money for her, and not to use it to finance shaky real estate operations in mosquito-infested countries.

A notary public is another arbitrator. When Bob receives a notarized document from Alice, he is convinced that Alice signed the document voluntarily and with her own hand. The notary can, if necessary, stand up in court and attest to that fact.

The concept of an arbitrator is as old as society. There have always been people—rulers, priests, and so on—who have the authority to act fairly. Arbitrators have a certain social role and position in our society; betraying the public trust would jeopardize that. Lawyers who play games with escrow accounts face almost-certain disbarment, for example. This picture of trust doesn't always exist in the real world, but it's the ideal.

This ideal can translate to the computer world, but there are several problems with computer arbitrators:

— It is easier to find and trust a neutral third party if you know who the party is and can see his face. Two parties suspicious of each other are also likely to be suspicious of a faceless arbitrator somewhere else on the network.

— The computer network must bear the cost of maintaining an arbitrator. We all know what lawyers charge; who wants to bear that kind of network overhead?

— There is a delay inherent in any arbitrated protocol.

— The arbitrator must deal with every transaction; he is a bottleneck in large-scale implementations of any protocol. Increasing the number of arbitrators in the implementation can mitigate this problem, but that increases the cost.

— Since everyone on the network must trust the arbitrator, he represents a vulnerable point for anyone trying to subvert the network.

Even so, arbitrators still have a role to play. In protocols using a trusted arbitrator, the part will be played by Trent.

Adjudicated Protocols

Because of the high cost of hiring arbitrators, arbitrated protocols can be subdivided into two lower-level **subprotocols**. One is a nonarbitrated subprotocol, executed every time parties want to complete the protocol. The other is an arbitrated subprotocol, executed only in exceptional circumstances—when there is a dispute. This special type of arbitrator is called an **adjudicator** (see Figure 2.1b).

An adjudicator is also a disinterested and trusted third party. Unlike an arbitrator, he is not directly involved in every protocol. The adjudicator is called in only to determine whether a protocol was performed fairly.

Judges are professional adjudicators. Unlike a notary public, a judge is brought in only if there is a dispute. Alice and Bob can enter into a contract without a judge. A judge never sees the contract until one of them hauls the other into court.

This contract-signing protocol can be formalized in this way:

Nonarbitrated subprotocol (executed every time):

(1) Alice and Bob negotiate the terms of the contract.
(2) Alice signs the contract.
(3) Bob signs the contract.

Adjudicated subprotocol (executed only in case of a dispute):

(4) Alice and Bob appear before a judge.
(5) Alice presents her evidence.
(6) Bob presents his evidence.
(7) The judge rules on the evidence.

The difference between an adjudicator and an arbitrator (as used in this book) is that the adjudicator is not always necessary. In a dispute, a judge is called in to adjudicate. If there is no dispute, using a judge is unnecessary.

There are adjudicated computer protocols. These protocols rely on the parties to be honest; but if someone suspects cheating, a body of data exists so that a trusted third party could determine if someone cheated. In a good adjudicated protocol, the adjudicator could also determine the cheater's identity. Instead of preventing cheating, adjudicated protocols detect cheating. The inevitability of detection acts as a preventive and discourages cheating.

Self-Enforcing Protocols

A **self-enforcing protocol** is the best type of protocol. The protocol itself guarantees fairness (see Figure 2.1c). No arbitrator is required to complete the protocol. No adjudicator is required to resolve disputes. The protocol is constructed so that there cannot be any disputes. If one of the parties tries to cheat, the other party immediately detects the cheating and the protocol stops. Whatever the cheating party hoped would happen by cheating, doesn't happen.

In the best of all possible worlds, every protocol would be self-enforcing. Unfortunately, there is not a self-enforcing protocol for every situation.

Attacks against Protocols

Cryptographic attacks can be directed against the cryptographic algorithms used in protocols, against the cryptographic techniques used to implement the algorithms and protocols, or against the protocols themselves. Since this section of the book discusses protocols, I will assume that the cryptographic algorithms and techniques are secure. I will only examine attacks against the protocols.

People can try various ways to attack a protocol. Someone not involved in the protocol can eavesdrop on some or all of the protocol. This is called a **passive attack**, because the attacker does not affect the protocol. All he can do is observe the protocol and attempt to gain information. This kind of attack corresponds to a ciphertext-only attack, as discussed in Section 1.1. Since passive attacks are difficult to detect, protocols try to prevent passive attacks rather than detect them. In these protocols, the part of the eavesdropper will be played by Eve.

Alternatively, an attacker could try to alter the protocol to his own advantage. He could pretend to be someone else, introduce new messages in the protocol, delete existing messages, substitute one message for another, replay old messages, interrupt a communications channel, or alter stored information in a computer. These are called **active attacks**, because they require active intervention. The form of these attacks depends on the network.

Passive attackers try to gain information about the parties involved in the protocol. They collect messages passing among various parties and attempt to cryptanalyze them. Active attacks, on the other hand, can have much more diverse objectives. The attacker could be interested in obtaining information, degrading system performance, corrupting existing information, or gaining unauthorized access to resources.

Active attacks are much more serious, especially in protocols in which the different parties don't necessarily trust one another. The attacker does not have to be a complete outsider. He could be a legitimate system user. He could be the system administrator. There could even be many active attackers working together. Here, the part of the malicious active attacker will be played by Mallory.

It is also possible that the attacker could be one of the parties involved in the protocol. He may lie during the protocol or not follow the protocol at all. This type of attacker is called a **cheater**. **Passive cheaters** follow the protocol, but try to obtain more information than the protocol intends them to. **Active cheaters** disrupt the protocol in progress in an attempt to cheat.

It is very difficult to maintain a protocol's security if most of the parties involved are active cheaters, but sometimes it is possible for legitimate parties to detect that active cheating is going on. Certainly, protocols should be secure against passive cheating.

2.2 COMMUNICATIONS USING SYMMETRIC CRYPTOGRAPHY

How do two parties communicate securely? They encrypt their communications, of course. The complete protocol is more complicated than that. Let's look at what must happen for Alice to send an encrypted message to Bob.

(1) Alice and Bob agree on a cryptosystem.
(2) Alice and Bob agree on a key.
(3) Alice takes her plaintext message and encrypts it using the encryption algorithm and the key. This creates a ciphertext message.
(4) Alice sends the ciphertext message to Bob.
(5) Bob decrypts the ciphertext message with the same algorithm and key and reads it.

What can Eve, sitting between Alice and Bob, learn from listening in on this protocol? If all she hears is the transmission in step (4), she must try to cryptanalyze the ciphertext. This passive attack is a ciphertext-only attack; we have algorithms that are resistant (as far as we know) to whatever computing power Eve could realistically bring to bear on the problem.

Eve isn't stupid, though. She also wants to listen in on steps (1) and (2). Then, she would know the algorithm and the key—just as well as Bob. When the message comes across the communications channel in step (4), all she has to do is decrypt it herself.

A good cryptosystem is one in which all the security is inherent in knowledge of the key and none is inherent in knowledge of the algorithm. This is why key management is so important in cryptography. With a symmetric algorithm, Alice and Bob can perform step (1) in public, but they must perform step (2) in secret. The key must remain secret before, during, and after the protocol—as long as the message must remain secret—otherwise the message will no longer be secure. (Public-key cryptography solves this problem another way, and will be discussed in Section 2.5.)

Mallory, an active attacker, could do a few other things. He could attempt to break the communications path in step (4), ensuring that Alice could not talk to Bob at all. Mallory could also intercept Alice's messages and substitute his own. If he knew the key (by intercepting the communication in step (2), or by breaking the cryptosystem), he could encrypt his own message and send it to Bob in place of the intercepted message. Bob would have no way of knowing that the message had not come from Alice. If Mallory didn't know the key, he could only create a replacement message that would decrypt to gibberish. Bob, thinking the message came from Alice, might conclude that either the network or Alice had some serious problems.

What about Alice? What can she do to disrupt the protocol? She can give a copy of the key to Eve. Now Eve can read whatever Bob says. She can reprint his words in *The New York Times.* Although serious, this is not a problem with the protocol. There is nothing to stop Alice from giving Eve a copy of the plaintext at any point during the protocol. Of course, Bob could also do anything that Alice could. This protocol assumes that Alice and Bob trust each other.

In summary, symmetric cryptosystems have the following problems:

- Keys must be distributed in secret. They are as valuable as all the messages they encrypt, since knowledge of the key gives knowledge of all the messages. For encryption systems that span the world, this can be a daunting task. Often couriers hand-carry keys to their destinations.
- If a key is compromised (stolen, guessed, extorted, bribed, etc.), then Eve can decrypt all message traffic encrypted with that key. She can also pretend to be one of the parties and produce false messages to fool the other party.
- Assuming a separate key is used for each pair of users in a network, the total number of keys increases rapidly as the number of users increases. A network of n users requires $n(n-1)/2$ keys. For example, 10 users require 45 different keys to talk with one another and 100 users require 4950 keys. This problem can be minimized by keeping the number of users small, but that is not always possible.

2.3 ONE-WAY FUNCTIONS

The notion of a **one-way function** is central to public-key cryptography. While not protocols in themselves, one-way functions are a fundamental building block for most of the protocols discussed in this book.

One-way functions are relatively easy to compute, but significantly harder to reverse. That is, given x it is easy to compute $f(x)$, but given $f(x)$ it is hard to compute x. In this context, "hard" is defined as something like: It would take millions of years to compute x from $f(x)$, even if all the computers in the world were assigned to the problem.

Breaking a plate is a good example of a one-way function. It is easy to smash a plate into a thousand tiny pieces. However, it's not easy to put all of those tiny pieces back together into a plate.

This sounds good, but it's a lot of smoke and mirrors. If we are being strictly mathematical, we have no proof that one-way functions exist, nor any real evidence that they can be constructed [230,530,600,661]. Even so, many functions look and smell one-way: We can compute them efficiently and, as of yet, know of no easy way to reverse them. For example, in a finite field x^2 is easy to compute, but $x^{1/2}$ is much harder. For the rest of this section, I'm going to pretend that there are one-way functions. I'll talk more about this in Section 11.2.

So, what good are one-way functions? We can't use them for encryption as is. A message encrypted with the one-way function isn't useful; no one could decrypt it. (Exercise: Write a message on a plate, smash the plate into tiny bits, and then give the bits to a friend. Ask your friend to read the message. Observe how impressed he is with the one-way function.) For public-key cryptography, we need something else (although there are cryptographic applications for one-way functions—see Section 3.2).

A **trapdoor one-way function** is a special type of one-way function, one with a secret trapdoor. It is easy to compute in one direction and hard to compute in the other direction. But, if you know the secret, you can easily compute the function in the other direction. That is, it is easy to compute $f(x)$ given x, and hard to compute x given $f(x)$. However, there is some secret information, y, such that given $f(x)$ and y it is easy to compute x.

Taking a watch apart is a good example of a trap-door one-way function. It is easy to disassemble a watch into hundreds of minuscule pieces. It is very difficult to put those tiny pieces back together into a working watch. However, with the secret information—the assembly instructions of the watch—it is much easier to put the watch back together.

2.4 ONE-WAY HASH FUNCTIONS

A **one-way hash function** has many names: compression function, contraction function, message digest, fingerprint, cryptographic checksum, message integrity check (MIC), and manipulation detection code (MDC). Whatever you call it, it is central to modern cryptography. One-way hash functions are another building block for many protocols.

Hash functions have been used in computer science for a long time. A hash function is a function, mathematical or otherwise, that takes a variable-length input string (called a **pre-image**) and converts it to a fixed-length (generally smaller) output string (called a **hash value**). A simple hash function would be a function that takes pre-image and returns a byte consisting of the XOR of all the input bytes.

The point here is to fingerprint the pre-image: to produce a value that indicates whether a candidate pre-image is likely to be the same as the real pre-image. Because hash functions are typically many-to-one, we cannot use them to determine with certainty that the two strings are equal, but we can use them to get a reasonable assurance of accuracy.

A one-way hash function is a hash function that works in one direction: It is easy to compute a hash value from pre-image, but it is hard to generate a pre-image that hashes to a particular value. The hash function previously mentioned is not one-way: Given a particular byte value, it is trivial to generate a string of bytes whose XOR is that value. You can't do that with a one-way hash function. A good one-way hash function is also **collision-free**: It is hard to generate two pre-images with the same hash value.

The hash function is public; there's no secrecy to the process. The security of a one-way hash function is its one-wayness. The output is not dependent on the input in any discernible way. A single bit change in the pre-image changes, on the average, half of the bits in the hash value. Given a hash value, it is computationally unfeasible to find a pre-image that hashes to that value.

Think of it as a way of fingerprinting files. If you want to verify that someone has a particular file (that you also have), but you don't want him to send it to you, then ask him for the hash value. If he sends you the correct hash value, then it is almost certain that he has that file. This is particularly useful in financial transactions, where you don't want a withdrawal of $100 to turn into a withdrawal of $1000 somewhere in the network. Normally, you would use a one-way hash function without a key, so that anyone can verify the hash. If you want only the recipient to be able to verify the hash, then read the next section.

Message Authentication Codes

A **message authentication code** (MAC), also known as a data authentication code (DAC), is a one-way hash function with the addition of a secret key (see Section 18.14). The hash value is a function of both the pre-image and the key. The theory is exactly the same as hash functions, except only someone with the key can verify the hash value. You can create a MAC out of a hash function or a block encryption algorithm; there are also dedicated MACs.

2.5 COMMUNICATIONS USING PUBLIC-KEY CRYPTOGRAPHY

Think of a symmetric algorithm as a safe. The key is the combination. Someone with the combination can open the safe, put a document inside, and close it again. Someone else with the combination can open the safe and take the document out. Anyone without the combination is forced to learn safecracking.

In 1976, Whitfield Diffie and Martin Hellman changed that paradigm of cryptography forever [496]. (The NSA has claimed knowledge of the concept as early as 1966, but has offered no proof.) They described **public-key cryptography**. They used two different keys—one public and the other private. It is computationally hard to deduce the private key from the public key. Anyone with the public key can encrypt a message but not decrypt it. Only the person with the private key can decrypt the message. It is as if someone turned the cryptographic safe into a mailbox. Putting mail in the mailbox is analogous to encrypting with the public key; anyone can do it. Just open the slot and drop it in. Getting mail out of a mailbox is analogous to decrypting with the private key. Generally it's hard; you need welding torches. However, if you have the secret (the physical key to the mailbox), it's easy to get mail out of a mailbox.

Mathematically, the process is based on the trap-door one-way functions previously discussed. Encryption is the easy direction. Instructions for encryption are the public key; anyone can encrypt a message. Decryption is the hard direction. It's made hard enough that people with Cray computers and thousands (even millions) of years couldn't decrypt the message without the secret. The secret, or trapdoor, is the private key. With that secret, decryption is as easy as encryption.

This is how Alice can send a message to Bob using public-key cryptography:

(1) Alice and Bob agree on a public-key cryptosystem.

(2) Bob sends Alice his public key.
(3) Alice encrypts her message using Bob's public key and sends it to Bob.
(4) Bob decrypts Alice's message using his private key.

Notice how public-key cryptography solves the key-management problem with symmetric cryptosystems. Before, Alice and Bob had to agree on a key in secret. Alice could choose one at random, but she still had to get it to Bob. She could hand it to him sometime beforehand, but that requires foresight. She could send it to him by secure courier, but that takes time. Public-key cryptography makes it easy. With no prior arrangements, Alice can send a secure message to Bob. Eve, listening in on the entire exchange, has Bob's public key and a message encrypted in that key, but cannot recover either Bob's private key or the message.

More commonly, a network of users agrees on a public-key cryptosystem. Every user has his or her own public key and private key, and the public keys are all published in a database somewhere. Now the protocol is even easier:

(1) Alice gets Bob's public key from the database.
(2) Alice encrypts her message using Bob's public key and sends it to Bob.
(3) Bob then decrypts Alice's message using his private key.

In the first protocol, Bob had to send Alice his public key before she could send him a message. The second protocol is more like traditional mail. Bob is not involved in the protocol until he wants to read his message.

Hybrid Cryptosystems

The first public-key algorithms became public at the same time that DES was being discussed as a proposed standard. This resulted in some partisan politics in the cryptographic community. As Diffie described it [494]:

> The excitement public key cryptosystems provoked in the popular and scientific press was not matched by corresponding acceptance in the cryptographic establishment, however. In the same year that public key cryptography was discovered, the National Security Agency (NSA), proposed a conventional cryptographic system, designed by International Business Machines (IBM), as a federal *Data Encryption Standard* (DES). Marty Hellman and I criticized the proposal on the ground that its key was too small, but manufacturers were gearing up to support the proposed standard and our criticism was seen by many as an attempt to disrupt the standards-making process to the advantage of our own work. Public key cryptography in its turn was attacked, in sales literature [1125] and technical papers [849,1159] alike, more as though it were a competing product than a recent research discovery. This, however, did not deter the NSA from claiming its share of the credit. Its director, in the words of the *Encyclopedia Britannica* [1461], pointed out that "two-key cryptography had been discovered at the agency a decade earlier," although no evidence for this claim was ever offered publicly.

In the real world, public-key algorithms are not a substitute for symmetric algorithms. They are not used to encrypt messages; they are used to encrypt keys. There are two reasons for this:

1. Public-key algorithms are slow. Symmetric algorithms are generally at least 1000 times faster than public-key algorithms. Yes, computers are getting faster and faster, and in 15 years computers will be able to do public-key cryptography at speeds comparable to symmetric cryptography today. But bandwidth requirements are also increasing, and there will always be the need to encrypt data faster than public-key cryptography can manage.
2. Public-key cryptosystems are vulnerable to chosen-plaintext attacks. If $C = E(P)$, when P is one plaintext out of a set of n possible plaintexts, then a cryptanalyst only has to encrypt all n possible plaintexts and compare the results with C (remember, the encryption key is public). He won't be able to recover the decryption key this way, but he will be able to determine P.

A chosen-plaintext attack can be particularly effective if there are relatively few possible encrypted messages. For example, if P were a dollar amount less than $1,000,000, this attack would work; the cryptanalyst tries all million possible dollar amounts. (Probabilistic encryption solves the problem; see Section 23.15.) Even if P is not as well-defined, this attack can be very effective. Simply knowing that a ciphertext does not correspond to a particular plaintext can be useful information. Symmetric cryptosystems are not vulnerable to this attack because a cryptanalyst cannot perform trial encryptions with an unknown key.

In most practical implementations public-key cryptography is used to secure and distribute **session keys**; those session keys are used with symmetric algorithms to secure message traffic [879]. This is sometimes called a **hybrid cryptosystem**.

(1) Bob sends Alice his public key.

(2) Alice generates a random session key, K, encrypts it using Bob's public key, and sends it to Bob.

$E_B(K)$

(3) Bob decrypts Alice's message using his private key to recover the session key.

$D_B(E_B(K)) = K$

(4) Both of them encrypt their communications using the same session key.

Using public-key cryptography for key distribution solves a very important key-management problem. With symmetric cryptography, the data encryption key sits around until it is used. If Eve ever gets her hands on it, she can decrypt messages encrypted with it. With the previous protocol, the session key is created when it is needed to encrypt communications and destroyed when it is no longer needed. This drastically reduces the risk of compromising the session key. Of course, the private key is vulnerable to compromise, but it is at less risk because it is only used once per communication to encrypt a session key. This is further discussed in Section 3.1.

Merkle's Puzzles

Ralph Merkle invented the first construction of public-key cryptography. In 1974 he registered for a course in computer security at the University of California, Berkeley, taught by Lance Hoffman. His term paper topic, submitted early in the term, addressed the problem of "Secure Communication over Insecure Channels" [1064]. Hoffman could not understand Merkle's proposal and eventually Merkle dropped the course. He continued to work on the problem, despite continuing failure to make his results understood.

Merkle's technique was based on "puzzles" that were easier to solve for the sender and receiver than for an eavesdropper. Here's how Alice sends an encrypted message to Bob without first having to exchange a key with him.

(1) Bob generates 2^{20}, or about a million, messages of the form: "This is puzzle number *x*. This is the secret key number *y*," where *x* is a random number and *y* is a random secret key. Both *x* and *y* are different for each message. Using a symmetric algorithm, he encrypts each message with a different 20-bit key and sends them all to Alice.

(2) Alice chooses one message at random and performs a brute-force attack to recover the plaintext. This is a large, but not impossible, amount of work.

(3) Alice encrypts her secret message with the key she recovered and some symmetric algorithm, and sends it to Bob along with *x*.

(4) Bob knows which secret key *y* he encrypts in message *x*, so he can decrypt the message.

Eve can break this system, but she has to do far more work than either Alice or Bob. To recover the message in step (3), she has to perform a brute-force attack against each of Bob's 2^{20} messages in step (1); this attack has a complexity of 2^{40}. The *x* values won't help Eve either; they were assigned randomly in step (1). In general, Eve has to expend approximately the square of the effort that Alice expends.

This n to n^2 advantage is small by cryptographic standards, but in some circumstances it may be enough. If Alice and Bob can try ten thousand keys per second, it will take them a minute each to perform their steps and another minute to communicate the puzzles from Bob to Alice on a 1.544 MB link. If Eve had comparable computing facilities, it would take her about a year to break the system. Other algorithms are even harder to break.

2.6 DIGITAL SIGNATURES

Handwritten signatures have long been used as proof of authorship of, or at least agreement with, the contents of a document. What is it about a signature that is so compelling [1392]?

1. The signature is authentic. The signature convinces the document's recipient that the signer deliberately signed the document.
2. The signature is unforgeable. The signature is proof that the signer, and no one else, deliberately signed the document.
3. The signature is not reusable. The signature is part of the document; an unscrupulous person cannot move the signature to a different document.
4. The signed document is unalterable. After the document is signed, it cannot be altered.
5. The signature cannot be repudiated. The signature and the document are physical things. The signer cannot later claim that he or she didn't sign it.

In reality, none of these statements about signatures is completely true. Signatures can be forged, signatures can be lifted from one piece of paper and moved to another, and documents can be altered after signing. However, we are willing to live with these problems because of the difficulty in cheating and the risk of detection.

We would like to do this sort of thing on computers, but there are problems. First, computer files are trivial to copy. Even if a person's signature were difficult to forge (a graphical image of a written signature, for example), it would be easy to cut and paste a valid signature from one document to another document. The mere presence of such a signature means nothing. Second, computer files are easy to modify after they are signed, without leaving any evidence of modification.

Signing Documents with Symmetric Cryptosystems and an Arbitrator

Alice wants to sign a digital message and send it to Bob. With the help of Trent and a symmetric cryptosystem, she can.

Trent is a powerful, trusted arbitrator. He can communicate with both Alice and Bob (and everyone else who may want to sign a digital document). He shares a secret key, K_A, with Alice, and a different secret key, K_B, with Bob. These keys have been established long before the protocol begins and can be reused multiple times for multiple signings.

(1) Alice encrypts her message to Bob with K_A and sends it to Trent.
(2) Trent decrypts the message with K_A.
(3) Trent takes the decrypted message and a statement that he has received this message from Alice, and encrypts the whole bundle with K_B.
(4) Trent sends the encrypted bundle to Bob.
(5) Bob decrypts the bundle with K_B. He can now read both the message and Trent's certification that Alice sent it.

How does Trent know that the message is from Alice and not from some imposter? He infers it from the message's encryption. Since only he and Alice share their secret key, only Alice could encrypt a message using it.

Is this as good as a paper signature? Let's look at the characteristics we want:

1. This signature is authentic. Trent is a trusted arbitrator and Trent knows that the message came from Alice. Trent's certification serves as proof to Bob.
2. This signature is unforgeable. Only Alice (and Trent, but everyone trusts him) knows K_A, so only Alice could have sent Trent a message encrypted with K_A. If someone tried to impersonate Alice, Trent would have immediately realized this in step (2) and would not certify its authenticity.
3. This signature is not reusable. If Bob tried to take Trent's certification and attach it to another message, Alice would cry foul. An arbitrator (it could be Trent or it could be a completely different arbitrator with access to the same information) would ask Bob to produce both the message and Alice's encrypted message. The arbitrator would then encrypt the message with K_A and see that it did not match the encrypted message that Bob gave him. Bob, of course, could not produce an encrypted message that matches because he does not know K_A.
4. The signed document is unalterable. Were Bob to try to alter the document after receipt, Trent could prove foul play in exactly the same manner just described.
5. The signature cannot be repudiated. Even if Alice later claims that she never sent the message, Trent's certification says otherwise. Remember, Trent is trusted by everyone; what he says is true.

If Bob wants to show Carol a document signed by Alice, he can't reveal his secret key to her. He has to go through Trent again:

(1) Bob takes the message and Trent's statement that the message came from Alice, encrypts them with K_B, and sends them back to Trent.
(2) Trent decrypts the bundle with K_B.
(3) Trent checks his database and confirms that the original message came from Alice.
(4) Trent re-encrypts the bundle with the secret key he shares with Carol, K_C, and sends it to Carol.
(5) Carol decrypts the bundle with K_C. She can now read both the message and Trent's certification that Alice sent it.

These protocols work, but they're time-consuming for Trent. He must spend his days decrypting and encrypting messages, acting as the intermediary between every pair of people who want to send signed documents to one another. He must keep a database of messages (although this can be avoided by sending the recipient a copy of the sender's encrypted message). He is a bottleneck in any communications system, even if he's a mindless software program.

Harder still is creating and maintaining someone like Trent, someone that everyone on the network trusts. Trent has to be infallible; if he makes even one mistake in a million signatures, no one is going to trust him. Trent has to be completely secure. If his database of secret keys ever got out or if someone managed to modify his programming, everyone's signatures would be completely useless. False documents purported to be signed years ago could appear. Chaos would result. Governments would collapse. Anarchy would reign. This might work in theory, but it doesn't work very well in practice.

Digital Signature Trees

Ralph Merkle proposed a digital signature scheme based on secret-key cryptography, producing an infinite number of one-time signatures using a tree structure [1067,1068]. The basic idea of this scheme is to place the root of the tree in some public file, thereby authenticating it. The root signs one message and authenticates its sub-nodes in the tree. Each of these nodes signs one message and authenticates its sub-nodes, and so on.

Signing Documents with Public-Key Cryptography

There are public-key algorithms that can be used for digital signatures. In some algorithms—RSA is an example (see Section 19.3)—either the public key or the private key can be used for encryption. Encrypt a document using your private key, and you have a secure digital signature. In other cases—DSA is an example (see Section 20.1)—there is a separate algorithm for digital signatures that cannot be used for encryption. This idea was first invented by Diffie and Hellman [496] and further expanded and elaborated on in other texts [1282,1328,1024,1283,426]. See [1099] for a good survey of the field.

The basic protocol is simple:

(1) Alice encrypts the document with her private key, thereby signing the document.
(2) Alice sends the signed document to Bob.
(3) Bob decrypts the document with Alice's public key, thereby verifying the signature.

This protocol is far better than the previous one. Trent is not needed to either sign or verify signatures. (He is needed to certify that Alice's public key is indeed her public key.) The parties do not even need Trent to resolve disputes: If Bob cannot perform step (3), then he knows the signature is not valid.

This protocol also satisfies the characteristics we're looking for:

1. The signature is authentic; when Bob verifies the message with Alice's public key, he knows that she signed it.
2. The signature is unforgeable; only Alice knows her private key.
3. The signature is not reusable; the signature is a function of the document and cannot be transferred to any other document.

4. The signed document is unalterable; if there is any alteration to the document, the signature can no longer be verified with Alice's public key.
5. The signature cannot be repudiated. Bob doesn't need Alice's help to verify her signature.

Signing Documents and Timestamps

Actually, Bob can cheat Alice in certain circumstances. He can reuse the document and signature together. This is no problem if Alice signed a contract (what's another copy of the same contract, more or less?), but it can be very exciting if Alice signed a digital check.

Let's say Alice sends Bob a signed digital check for $100. Bob takes the check to the bank, which verifies the signature and moves the money from one account to the other. Bob, who is an unscrupulous character, saves a copy of the digital check. The following week, he again takes it to the bank (or maybe to a different bank). The bank verifies the signature and moves the money from one account to the other. If Alice never balances her checkbook, Bob can keep this up for years.

Consequently, digital signatures often include timestamps. The date and time of the signature are attached to the message and signed along with the rest of the message. The bank stores this timestamp in a database. Now, when Bob tries to cash Alice's check a second time, the bank checks the timestamp against its database. Since the bank already cashed a check from Alice with the same timestamp, the bank calls the police. Bob then spends 15 years in Leavenworth prison reading up on cryptographic protocols.

Signing Documents with Public-Key Cryptography and One-Way Hash Functions

In practical implementations, public-key algorithms are often too inefficient to sign long documents. To save time, digital signature protocols are often implemented with one-way hash functions [432,433]. Instead of signing a document, Alice signs the hash of the document. In this protocol, both the one-way hash function and the digital signature algorithm are agreed upon beforehand.

(1) Alice produces a one-way hash of a document.
(2) Alice encrypts the hash with her private key, thereby signing the document.
(3) Alice sends the document and the signed hash to Bob.
(4) Bob produces a one-way hash of the document that Alice sent. He then, using the digital signature algorithm, decrypts the signed hash with Alice's public key. If the signed hash matches the hash he generated, the signature is valid.

Speed increases drastically and, since the chances of two different documents having the same 160-bit hash are only one in 2^{160}, anyone can safely equate a signature of the hash with a signature of the document. If a non-one-way hash function were used, it would be an easy matter to create multiple documents that hashed to the same value, so that anyone signing a particular document would be duped into signing a multitude of documents.

This protocol has other benefits. First, the signature can be kept separate from the document. Second, the recipient's storage requirements for the document and signature are much smaller. An archival system can use this type of protocol to verify the existence of documents without storing their contents. The central database could just store the hashes of files. It doesn't have to see the files at all; users submit their hashes to the database, and the database timestamps the submissions and stores them. If there is any disagreement in the future about who created a document and when, the database could resolve it by finding the hash in its files. This system has vast implications concerning privacy: Alice could copyright a document but still keep the document secret. Only if she wished to prove her copyright would she have to make the document public. (See Section 4.1).

Algorithms and Terminology

There are many digital signature algorithms. All of them are public-key algorithms with secret information to sign documents and public information to verify signatures. Sometimes the signing process is called **encrypting with a private key** and the verification process is called **decrypting with a public key**. This is misleading and is only true for one algorithm, RSA. And different algorithms have different implementations. For example, one-way hash functions and timestamps sometimes add extra steps to the process of signing and verifying. Many algorithms can be used for digital signatures, but not for encryption.

In general, I will refer to the signing and verifying processes without any details of the algorithms involved. Signing a message with private key K is:

$$S_K(M)$$

and verifying a signature with the corresponding public key is:

$$V_K(M)$$

The bit string attached to the document when signed (in the previous example, the one-way hash of the document encrypted with the private key) will be called the **digital signature**, or just the **signature**. The entire protocol, by which the receiver of a message is convinced of the identity of the sender and the integrity of the message, is called authentication. Further details on these protocols are in Section 3.2.

Multiple Signatures

How could Alice and Bob sign the same digital document? Without one-way hash functions, there are two options. One is that Alice and Bob sign separate copies of the document itself. The resultant message would be over twice the size of the original document. The second is that Alice signs the document first and then Bob signs Alice's signature. This works, but it is impossible to verify Alice's signature without also verifying Bob's.

With one-way hash functions, multiple signatures are easy:

(1) Alice signs the hash of the document.
(2) Bob signs the hash of the document.
(3) Bob sends his signature to Alice.
(4) Alice sends the document, her signature, and Bob's signature to Carol.
(5) Carol verifies both Alice's signature and Bob's signature.

Alice and Bob can do steps (1) and (2) either in parallel or in series. In step (5), Carol can verify one signature without having to verify the other.

Nonrepudiation and Digital Signatures

Alice can cheat with digital signatures and there's nothing that can be done about it. She can sign a document and then later claim that she did not. First, she signs the document normally. Then, she anonymously publishes her private key, conveniently loses it in a public place, or just pretends to do either one. Alice then claims that her signature has been compromised and that others are using it, pretending to be her. She disavows signing the document and any others that she signed using that private key. This is called repudiation.

Timestamps can limit the effects of this kind of cheating, but Alice can always claim that her key was compromised earlier. If Alice times things well, she can sign a document and then successfully claim that she didn't. This is why there is so much talk about private keys buried in tamper-resistant modules—so that Alice can't get at hers and abuse it.

Although nothing can be done about this possible abuse, one can take steps to guarantee that old signatures are not invalidated by actions taken in disputing new ones. (For example, Alice could "lose" her key to keep from paying Bob for the junk car he sold her yesterday and, in the process, invalidate her bank account.) The solution is for the receiver of a signed document to have it timestamped [453].

The general protocol is given in [28]:

(1) Alice signs a message.
(2) Alice generates a header containing some identifying information. She concatenates the header with the signed message, signs that, and sends it to Trent.
(3) Trent verifies the outside signature and confirms the identifying information. He adds a timestamp to Alice's signed message and the identifying information. Then he signs it all and sends it to both Alice and Bob.
(4) Bob verifies Trent's signature, the identifying information, and Alice's signature.
(5) Alice verifies the message Trent sent to Bob. If she did not originate the message, she speaks up quickly.

Another scheme uses Trent after the fact [209]. After receiving a signed message, Bob can send a copy to Trent for verification. Trent can attest to the validity of Alice's signature.

Applications of Digital Signatures

One of the earliest proposed applications of digital signatures was to facilitate the verification of nuclear test ban treaties [1454,1467]. The United States and the Soviet Union (anyone remember the Soviet Union?) permitted each other to put seismometers on the other's soil to monitor nuclear tests. The problem was that each country needed to assure itself that the host nation was not tampering with the data from the monitoring nation's seismometers. Simultaneously, the host nation needed to assure itself that the monitor was sending only the specific information needed for monitoring.

Conventional authentication techniques can solve the first problem, but only digital signatures can solve both problems. The host nation can read, but not alter, data from the seismometer, and the monitoring nation knows that the data has not been tampered with.

2.7 DIGITAL SIGNATURES WITH ENCRYPTION

By combining digital signatures with public-key cryptography, we develop a protocol that combines the security of encryption with the authenticity of digital signatures. Think of a letter from your mother: The signature provides proof of authorship and the envelope provides privacy.

(1) Alice signs the message with her private key.
$S_A(M)$
(2) Alice encrypts the signed message with Bob's public key and sends it to Bob.
$E_B(S_A(M))$
(3) Bob decrypts the message with his private key.
$D_B(E_B(S_A(M))) = S_A(M)$
(4) Bob verifies with Alice's public key and recovers the message.
$V_A(S_A(M)) = M$

Signing before encrypting seems natural. When Alice writes a letter, she signs it and then puts it in an envelope. If she put the letter in the envelope unsigned and then signed the envelope, then Bob might worry if the letter hadn't been covertly replaced. If Bob showed to Carol Alice's letter and envelope, Carol might accuse Bob of lying about which letter arrived in which envelope.

In electronic correspondence as well, signing before encrypting is a prudent practice [48]. Not only is it more secure—an adversary can't remove a signature from an encrypted message and add his own—but there are legal considerations: If the text to be signed is not visible to the signer when he affixes his signature, then the signature may have little legal force [1312]. And there are some cryptanalytic attacks against this technique with RSA signatures (see Section 19.3).

There's no reason Alice has to use the same public-key/private-key key pair for encrypting and signing. She can have two key pairs: one for encryption and the other for signatures. Separation has its advantages: she can surrender her encryption key to the police without compromising her signature, one key can be escrowed (see Section 4.13) without affecting the other, and the keys can have different sizes and can expire at different times.

Of course, timestamps should be used with this protocol to prevent reuse of messages. Timestamps can also protect against other potential pitfalls, such as the one described below.

Resending the Message as a Receipt

Consider an implementation of this protocol, with the additional feature of confirmation messages. Whenever Bob receives a message, he returns it as a confirmation of receipt.

(1) Alice signs a message with her private key, encrypts it with Bob's public key, and sends it to Bob.

$E_B(S_A(M))$

(2) Bob decrypts the message with his private key and verifies the signature with Alice's public key, thereby verifying that Alice signed the message and recovering the message.

$V_A(D_B(E_B(S_A(M)))) = M$

(3) Bob signs the message with his private key, encrypts it with Alice's public key, and sends it back to Alice.

$E_A(S_B(M))$

(4) Alice decrypts the message with her private key and verifies the signature with Bob's public key. If the resultant message is the same one she sent to Bob, she knows that Bob received the message accurately.

If the same algorithm is used for both encryption and digital-signature verification there is a possible attack [506]. In these cases, the digital signature operation is the inverse of the encryption operation: $V_X = E_X$ and $S_X = D_X$.

Assume that Mallory is a legitimate system user with his own public and private key. Now, let's watch as he reads Bob's mail. First, he records Alice's message to Bob in step (1). Then, at some later time, he sends that message to Bob, claiming that it came from him (Mallory). Bob thinks that it is a legitimate message from Mallory, so he decrypts the message with his private key and then tries to verify Mallory's signature by decrypting it with Mallory's public key. The resultant message, which is pure gibberish, is:

$$E_M(D_B(E_B(D_A(M)))) = E_M(D_A(M))$$

Even so, Bob goes on with the protocol and sends Mallory a receipt:

$$E_M(D_B(E_M(D_A(M))))$$

Now, all Mallory has to do is decrypt the message with his private key, encrypt it with Bob's public key, decrypt it again with his private key, and encrypt it with Alice's public key. *Voilà!* Mallory has *M*.

It is not unreasonable to imagine that Bob may automatically send Mallory a receipt. This protocol may be embedded in his communications software, for example, and send receipts automatically. It is this willingness to acknowledge the receipt of gibberish that creates the insecurity. If Bob checked the message for comprehensibility before sending a receipt, he could avoid this security problem.

There are enhancements to this attack that allow Mallory to send Bob a different message from the one he eavesdropped on. Never sign arbitrary messages from other people or decrypt arbitrary messages and give the results to other people.

Foiling the Resend Attack

The attack just described works because the encrypting operation is the same as the signature-verifying operation and the decryption operation is the same as the signature operation. A secure protocol would use even a slightly different operation for encryption and digital signatures. Using different keys for each operation solves the problem, as does using different algorithms for each operation; as do timestamps, which make the incoming message and the outgoing message different; as do digital signatures with one-way hash functions (see Section 2.6).

In general, then, the following protocol is secure as the public-key algorithm used:

(1) Alice signs a message.
(2) Alice encrypts the message and signature with Bob's public key (using a different encryption algorithm than for the signature) and sends it to Bob.
(3) Bob decrypts the message with his private key.
(4) Bob verifies Alice's signature.

Attacks against Public-Key Cryptography

In all these public-key cryptography protocols, I glossed over how Alice gets Bob's public key. Section 3.1 discusses this in detail, but it is worth mentioning here.

The easiest way to get someone's public key is from a secure database somewhere. The database has to be public, so that anyone can get anyone else's public key. The database also has to be protected from write-access by anyone except Trent; otherwise Mallory could substitute any public key for Bob's. After he did that, Bob couldn't read messages addressed to him, but Mallory could.

Even if the public keys are stored in a secure database, Mallory could still substitute one for another during transmission. To prevent this, Trent can sign each public key with his own private key. Trent, when used in this manner, is often known as a **Key Certification Authority** or **Key Distribution Center** (**KDC**). In practical implementations, the KDC signs a compound message consisting of the user's name, his public key, and any other important information about the user. This signed compound message is stored in the KDC's database. When Alice gets Bob's key, she verifies the KDC's signature to assure herself of the key's validity.

In the final analysis, this is not making things impossible for Mallory, only more difficult. Alice still has the KDC's public key stored somewhere. Mallory would have to substitute his own public key for that key, corrupt the database, and substitute his own keys for the valid keys (all signed with his private key as if he were the KDC), and then he's in business. But, even paper-based signatures can be forged if Mallory goes to enough trouble. Key exchange will be discussed in minute detail in Section 3.1.

2.8 RANDOM AND PSEUDO-RANDOM-SEQUENCE GENERATION

Why even bother with random-number generation in a book on cryptography? There's already a random-number generator built into most every compiler, a mere function call away. Why not use that? Unfortunately, those random-number generators are almost definitely not secure enough for cryptography, and probably not even very random. Most of them are embarrassingly bad.

Random-number generators are not random because they don't have to be. Most simple applications, like computer games, need so few random numbers that they hardly notice. However, cryptography is extremely sensitive to the properties of random-number generators. Use a poor random-number generator and you start getting weird correlations and strange results [1231,1238]. If you are depending on your random-number generator for security, weird correlations and strange results are the last things you want.

The problem is that a random-number generator doesn't produce a random sequence. It probably doesn't produce anything that looks even remotely like a random sequence. Of course, it is impossible to produce something truly random on a computer. Donald Knuth quotes John von Neumann as saying: "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin" [863]. Computers are deterministic beasts: Stuff goes in one end, completely predictable operations occur inside, and different stuff comes out the other end. Put the same stuff in on two separate occasions and the same stuff comes out both times. Put the same stuff into two identical computers, and the same stuff comes out of both of them. A computer can only be in a finite number of states (a large finite number, but a finite number nonetheless), and the stuff that comes out will always be a deterministic function of the stuff that went in and the computer's current state. That means that any random-number generator on a computer (at least, on a finite-state machine) is, by definition, periodic. Anything that is periodic is, by definition, predictable. And if something is predictable, it can't be random. A true random-number generator requires some random input; a computer can't provide that.

Pseudo-Random Sequences

The best a computer can produce is a **pseudo-random-sequence generator**. What's that? Many people have taken a stab at defining this formally, but I'll hand-wave here. A pseudo-random sequence is one that looks random. The sequence's period should be long enough so that a finite sequence of reasonable length—that is, one that is actually used—is not periodic. If you need a billion random bits, don't choose a sequence generator that repeats after only sixteen thousand bits. These relatively short nonperiodic subsequences should be as indistinguishable as possible from random sequences. For example, they should have about the same number of ones and zeros, about half the runs (sequences of the same bit) should be of length one, one quarter of length two, one eighth of length three, and so on. They should not be compressible. The distribution of run lengths for zeros and ones should be the same [643,863,99,1357]. These properties can be empirically measured and then compared to statistical expectations using a chi-square test.

For our purposes, a sequence generator is pseudo-random if it has this property:

1. It looks random. This means that it passes all the statistical tests of randomness that we can find. (Start with the ones in [863].)

A lot of effort has gone into producing good pseudo-random sequences on computer. Discussions of generators abound in the academic literature, along with various tests of randomness. All of these generators are periodic (there's no escaping that); but with potential periods of 2^{256} bits and higher, they can be used for the largest applications.

The problem is still those weird correlations and strange results. Every pseudo-random-sequence generator is going to produce them if you use them in a certain way. And that's what a cryptanalyst will use to attack the system.

Cryptographically Secure Pseudo-Random Sequences

Cryptographic applications demand much more of a pseudo-random-sequence generator than do most other applications. Cryptographic randomness doesn't mean just statistical randomness, although that's part of it. For a sequence to be **cryptographically secure pseudo-random**, it must also have this property:

2. It is unpredictable. It must be computationally infeasible to predict what the next random bit will be, given complete knowledge of the algorithm or hardware generating the sequence and all of the previous bits in the stream.

Cryptographically secure pseudo-random sequences should not be compressible . . . unless you know the key. The key is generally the seed used to set the initial state of the generator.

Like any cryptographic algorithm, cryptographically secure pseudo-random-sequence generators are subject to attack. Just as it is possible to break an encryption algorithm, it is possible to break a cryptographically secure pseudo-random-sequence generator. Making generators resistant to attack is what cryptography is all about.

Real Random Sequences

Now we're drifting into the domain of philosophers. Is there such a thing as randomness? What is a random sequence? How do you know if a sequence is random? Is "101110100" more random than "101010101"? Quantum mechanics tells us that there is honest-to-goodness randomness in the real world. But can we preserve that randomness in the deterministic world of computer chips and finite-state machines?

Philosophy aside, from our point of view a sequence generator is **real random** if it has this additional third property:

3. It cannot be reliably reproduced. If you run the sequence generator twice with the exact same input (at least as exact as humanly possible), you will get two completely unrelated random sequences.

The output of a generator satisfying these three properties will be good enough for a one-time pad, key generation, and any other cryptographic applications that require a truly random sequence generator. The difficulty is in determining whether a sequence is really random. If I repeatedly encrypt a string with DES and a given key, I will get a nice, random-looking output; you won't be able to tell that it's nonrandom unless you rent time on the NSA's DES cracker.

CHAPTER 8

Key Management

Alice and Bob have a secure communications system. They play mental poker, simultaneously sign contracts, even exchange digital cash. Their protocols are secure. Their algorithms are top-notch. Unfortunately, they buy their keys from Eve's "Keys-R-Us," whose slogan is "You can trust us: Security is the middle name of someone our ex-mother-in-law's travel agent met at the Kwik-E-Mart."

Eve doesn't have to break the algorithms. She doesn't have to rely on subtle flaws in the protocols. She can use their keys to read all of Alice's and Bob's message traffic without lifting a cryptanalytic finger.

In the real world, key management is the hardest part of cryptography. Designing secure cryptographic algorithms and protocols isn't easy, but you can rely on a large body of academic research. Keeping the keys secret is much harder.

Cryptanalysts often attack both symmetric and public-key cryptosystems through their key management. Why should Eve bother going through all the trouble of trying to break the cryptographic algorithm if she can recover the key because of sloppy key storage procedures? Why should she spend $10 million building a cryptanalysis machine if she can spend $1000 bribing a clerk? Spending a million dollars to buy a well-placed communications clerk in a diplomatic embassy can be a bargain. The Walkers sold U.S. Navy encryption keys to the Soviets for years. The CIA's director of counterintelligence went for less than $2 million, wife included. That's far cheaper than building massive cracking machines and hiring brilliant cryptanalysts. Eve can steal the keys. She can arrest or abduct someone who knows the keys. She can seduce someone and get the keys that way. (The Marines who guarded the U.S. Embassy in Moscow were not immune to that attack.) It's a whole lot easier to find flaws in people than it is to find them in cryptosystems.

Alice and Bob must protect their key to the same degree as all the data it encrypts. If a key isn't changed regularly, this can be an enormous amount of data. Unfortunately, many commercial products simply proclaim "We use DES" and forget about everything else. The results are not very impressive.

For example, the DiskLock program for Macintosh (version 2.1), sold at most software stores, claims the security of DES encryption. It encrypts files using DES. Its implementation of the DES algorithm is correct. However, DiskLock stores the DES key with the encrypted file. If you know where to look for the key, and want to read a file encrypted with DiskLock's DES, recover the key from the encrypted file and then decrypt the file. It doesn't matter that this program uses DES encryption—the implementation is completely insecure.

Further information on key management can be found in [457,98,1273,1225, 775,357]. The following sections discuss some of the issues and solutions.

8.1 GENERATING KEYS

The security of an algorithm rests in the key. If you're using a cryptographically weak process to generate keys, then your whole system is weak. Eve need not cryptanalyze your encryption algorithm; she can cryptanalyze your key generation algorithm.

Reduced Keyspaces

DES has a 56-bit key. Implemented properly, any 56-bit string can be the key; there are 2^{56} (10^{16}) possible keys. Norton Discreet for MS-DOS (versions 8.0 and earlier) only allows ASCII keys, forcing the high-order bit of each byte to be zero. The program also converts lowercase letters to uppercase (so the fifth bit of each byte is always the opposite of the sixth bit) and ignores the low-order bit of each byte, resulting in only 2^{40} possible keys. These poor key generation procedures have made its DES ten thousand times easier to break than a proper implementation.

Table 8.1 gives the number of possible keys with various constraints on the input strings. Table 8.2 gives the time required for an exhaustive search through all of those keys, given a million attempts per second. Remember, there is very little time differential between an exhaustive search for 8-byte keys and an exhaustive search of 4-, 5-, 6-, 7-, and 8-byte keys.

All specialized brute-force hardware and parallel implementations will work here. Testing a million keys per second (either with one machine or with multiple machines in parallel), it is feasible to crack lowercase-letter and lowercase-letter-

Table 8.1
Number of Possible Keys of Various Keyspaces

	4-Byte	5-Byte	6-Byte	7-Byte	8-Byte
Lowercase letters (26):	460,000	$1.2*10^7$	$3.1*10^8$	$8.0*10^9$	$2.1*10^{11}$
Lowercase letters and digits (36):	1,700,000	$6.0*10^7$	$2.2*10^9$	$7.8*10^{10}$	$2.8*10^{12}$
Alphanumeric characters (62):	$1.5*10^7$	$9.2*10^8$	$5.7*10^{10}$	$3.5*10^{12}$	$2.2*10^{14}$
Printable characters (95):	$8.1*10^7$	$7.7*10^9$	$7.4*10^{11}$	$7.0*10^{13}$	$6.6*10^{15}$
ASCII characters (128):	$2.7*10^8$	$3.4*10^{10}$	$4.4*10^{12}$	$5.6*10^{14}$	$7.2*10^{16}$
8-bit ASCII characters (256):	$4.3*10^9$	$1.1*10^{12}$	$2.8*10^{14}$	$7.2*10^{16}$	$1.8*10^{19}$

Table 8.2
Exhaustive Search of Various Keyspaces (assume one million attempts per second)

	4-Byte	5-Byte	6-Byte	7-Byte	8-Byte
Lowercase letters (26):	.5 seconds	12 seconds	5 minutes	2.2 hours	2.4 days
Lowercase letters and digits (36):	1.7 seconds	1 minute	36 minutes	22 hours	33 days
Alphanumeric characters (62):	15 seconds	15 minutes	16 hours	41 days	6.9 years
Printable characters (95):	1.4 minutes	2.1 hours	8.5 days	2.2 years	210 years
ASCII characters (128):	4.5 minutes	9.5 hours	51 days	18 years	2300 years
8-bit ASCII characters (256):	1.2 hours	13 days	8.9 years	2300 years	580,000 years

and-number keys up to 8 bytes long, alphanumeric-character keys up to 7 bytes long, printable character and ASCII-character keys up to 6 bytes long, and 8-bit-ASCII-character keys up to 5 bytes long.

And remember, computing power doubles every 18 months. If you expect your keys to stand up against brute-force attacks for 10 years, you'd better plan accordingly.

Poor Key Choices

When people choose their own keys, they generally choose poor ones. They're far more likely to choose "Barney" than "*9 (hH/A." This is not always due to poor security practices; "Barney" is easier to remember than "*9 (hH/A." The world's most secure algorithm won't help much if the users habitually choose their spouse's names for keys or write their keys on little pieces of paper in their wallets. A smart brute-force attack doesn't try all possible keys in numerical order; it tries the obvious keys first.

This is called a **dictionary attack**, because the attacker uses a dictionary of common keys. Daniel Klein was able to crack 40 percent of the passwords on the average computer using this system [847,848]. No, he didn't try one password after another, trying to login. He copied the encrypted password file and mounted the attack offline. Here's what he tried:

1. The user's name, initials, account name, and other relevant personal information as a possible password. All in all, up to 130 different passwords were tried based on this information. For an account name **klone** with a user named "Daniel V. Klein," some of the passwords that would be tried were: klone, klone0, klone1, klone123, dvk, dvkdvk, dklein, DKlein leinad, nielk, dvklein, danielk, DvkkvD, DANIEL-KLEIN, (klone), KleinD, and so on.
2. Words from various databases. These included lists of men's and women's names (some 16,000 in all); places (including variations so that "spain," "spanish," and "spaniard" would all be considered); names of famous people; cartoons and cartoon characters; titles, characters, and locations from films and science fiction stories; mythical creatures (garnered from *Bullfinch's Mythology* and dictionaries of mythical beasts); sports (includ-

ing team names, nicknames, and specialized terms); numbers (both as numerals—"2001," and written out—"twelve"); strings of letters and numbers ("a," "aa," "aaa," "aaaa," etc.); Chinese syllables (from the Pinyin Romanization of Chinese, an international standard system of writing Chinese on an English keyboard); the King James Bible; biological terms; colloquial and vulgar phrases (such as "fuckyou," "ibmsux," and "deadhead"); keyboard patterns (such as "qwerty," "asdf," and "zxcvbn"); abbreviations (such as "roygbiv"—the colors in the rainbow, and "ooottafagvah"—a mnemonic for remembering the 12 cranial nerves); machine names (acquired from */etc/hosts*); characters, plays, and locations from Shakespeare; common Yiddish words; the names of asteroids; and a collection of words from various technical papers Klein previously published. All told, more than 60,000 separate words were considered per user (with any inter- and intra-dictionary duplicates being discarded).

3. Variations on the words from step 2. This included making the first letter uppercase or a control character, making the entire word uppercase, reversing the word (with and without the aforementioned capitalization), changing the letter 'o' to the digit '0' (so that the word "scholar" would also be checked as "sch0lar"), changing the letter 'l' to the digit '1' (so that the word "scholar" would also be checked as "scho1ar"), and performing similar manipulation to change the letter 'z' into the digit '2', and the letter 's' into the digit '5'. Another test was to make the word into a plural (irrespective of whether the word was actually a noun), with enough intelligence built in so that "dress" became "dresses," "house" became "houses," and "daisy" became "daisies." Klein did not consider pluralization rules exclusively, though, so that "datum" forgivably became "datums" (not "data"), while "sphynx" became "sphynxs" (and not "sphynges"). Similarly, the suffixes "-ed," "-er," and "-ing" were added to transform words like "phase" into "phased," "phaser," and "phasing." These additional tests added another 1,000,000 words to the list of possible passwords that were tested for each user.
4. Various capitalization variations on the words from step 2 that were not considered in step 3. This included all single-letter capitalization variations (so that "michael" would also be checked as "mIchael," "miChael," "micHael," "michAel," etc.), double-letter capitalization variations ("MIchael," "MiChael," "MicHael," . . . , "mIChael," "mIcHael," etc.), triple-letter variations, etc. The single-letter variations added roughly another 400,000 words to be checked per user, while the double-letter variations added another 1,500,000 words. Three-letter variations would have added at least another 3,000,000 words per user had there been enough time to complete the tests. Tests of four-, five-, and six-letter variations were deemed to be impracticable without much more computational horsepower to carry them out.
5. Foreign language words on foreign users. The specific test that was performed was to try Chinese language passwords on users with Chinese

names. The Pinyin Romanization of Chinese syllables was used, combining syllables together into one-, two-, and three-syllable words. Because no tests were done to determine whether the words actually made sense, an exhaustive search was initiated. Since there are 298 Chinese syllables in the Pinyin system, there are 158,404 two-syllable words, and slightly more than 16,000,000 three-syllable words. A similar mode of attack could as easily be used with English, using rules for building pronounceable nonsense words.

6. Word pairs. The magnitude of an exhaustive test of this nature is staggering. To simplify the test, only words of three or four characters in length from */usr/dict/words* were used. Even so, the number of word pairs is about ten million.

A dictionary attack is much more powerful when it is used against a file of keys and not a single key. A single user may be smart enough to choose good keys. If a thousand people each choose their own key as a password to a computer system, the odds are excellent that at least one person will choose a key in the attacker's dictionary.

Random Keys

Good keys are random-bit strings generated by some automatic process. If the key is 64 bits long, every possible 64-bit key must be equally likely. Generate the key bits from either a reliably random source (see Section 17.14) or a cryptographically secure pseudo-random-bit generator (see Chapters 16 and 17.) If these automatic processes are unavailable, flip a coin or roll a die.

This is important, but don't get too caught up in arguing about whether random noise from audio sources is more random than random noise from radioactive decay. None of these random-noise sources will be perfect, but they will probably be good enough. It is important to use a good random-number generator for key generation, but it is far more important to use good encryption algorithms and key management procedures. If you are worried about the randomness of your keys, use the key-crunching technique described below.

Some encryption algorithms have weak keys: specific keys that are less secure than the other keys. I advise testing for these weak keys and generating a new one if you discover one. DES has only 16 weak keys out of 2^{56}, so the odds of generating any of these keys are incredibly small. It has been argued that a cryptanalyst would have no idea that a weak key is being used and therefore gains no advantage from their accidental use. It has also been argued that not using weak keys gives a cryptanalyst information. However, testing for the few weak keys is so easy that it seems imprudent not to do so.

Generating keys for public-key cryptography systems is harder, because often the keys must have certain mathematical properties (they may have to be prime, be a quadratic residue, etc.). Techniques for generating large random prime numbers are discussed in Section 11.5. The important thing to remember from a key management point of view is that the random seeds for those generators must be just that: random.

Generating a random key isn't always possible. Sometimes you need to remember your key. (See how long it takes you to remember 25e8 56f2 e8ba c820). If you have to generate an easy-to-remember key, make it obscure. The ideal would be something easy to remember, but difficult to guess. Here are some suggestions:

- Word pairs separated by a punctuation character, for example "turtle*moose" or "zorch!splat"
- Strings of letters that are an acronym of a longer phrase; for example, "Mein Luftkissenfahrzeug ist voller Aale!" generates the key "MLivA!"

Pass Phrases

A better solution is to use an entire phrase instead of a word, and to convert that phrase into a key. These phrases are called **pass phrases**. A technique called **key crunching** converts the easy-to-remember phrases into random keys. Use a one-way hash function to transform an arbitrary-length text string into a pseudo-random-bit string.

For example, the easy-to-remember text string:

```
My name is Ozymandias, king of kings. Look on my works, ye mighty, and despair.
```

might crunch into this 64-bit key:

```
e6c1 4398 5ae9 0a9b
```

Of course, it can be difficult to type an entire phrase into a computer with the echo turned off. Clever suggestions to solve this problem would be appreciated.

If the phrase is long enough, the resulting key will be random. Exactly what "long enough" means is open to interpretation. Information theory tells us that standard English has about 1.3 bits of information per character (see Section 11.1). For a 64-bit key, a pass phrase of about 49 characters, or 10 normal English words, should be sufficient. As a rule of thumb, figure that you need five words for each 4 bytes of key. That's a conservative assumption, since it doesn't take into account case, spacing, and punctuation.

This technique can even be used to generate private keys for public-key cryptography systems: The text string could be crunched into a random seed, and that seed could be fed into a deterministic system that generates public-key/private-key key pairs.

If you are choosing a pass phrase, choose something unique and easy-to-remember. Don't choose phrases from literature—the example from "Ozymandias" is a bad one. Both the complete works of Shakespeare and the dialogue from *Star Wars* are available on-line and can be used in a dictionary attack. Choose something obscure, but personal. Include punctuation and capitalization; if you can, include numbers and non-alphanumeric symbols. Poor or improper English, or even a foreign language, makes the pass phrase less susceptible to a dictionary attack. One suggestion is to use a phrase that is "shocking nonsense": something offensive enough that you are likely to remember and unlikely to write down.

Despite everything written here, obscurity is no substitute for true randomness. The best keys are random keys, difficult as they are to remember.

X9.17 Key Generation

The ANSI X9.17 standard specifies a method of key generation (see Figure 8.1) [55]. This does not generate easy-to-remember keys; it is more suitable for generating session keys or pseudo-random numbers within a system. The cryptographic algorithm used to generate keys is triple-DES, but it could just as easily be any algorithm.

Let $E_K(X)$ be triple-DES encryption of X with key K. This is a special key reserved for secret key generation. V_0 is a secret 64-bit seed. T is a timestamp. To generate the random key R_i, calculate:

$$R_i = E_K(E_K(T_i) \oplus V_i)$$

To generate V_{i+1}, calculate:

$$V_{i+1} = E_K(E_K(T_i) \oplus R_i)$$

To turn R_i into a DES key, simply adjust every eighth bit for parity. If you need a 64-bit key, use it as is. If you need a 128-bit key, generate a pair of keys and concatenate them together.

DoD Key Generation

The U.S. Department of Defense recommends using DES in OFB mode (see Section 9.8) to generate random keys [1144]. Generate a DES key from system interrupt vectors, system status registers, and system counters. Generate an initialization vector from the system clock, system ID, and date and time. For the plaintext, use an externally generated 64-bit quantity: eight characters typed in by a system administrator, for example. Use the output as your key.

8.2 NONLINEAR KEYSPACES

Imagine that you are a military cryptography organization, building a piece of cryptography equipment for your troops. You want to use a secure algorithm, but you are

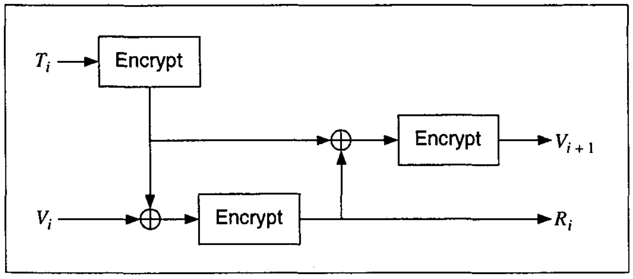


Figure 8.1 ANSI X9.17 key generation.

worried about the equipment falling into enemy hands. The last thing you want is for your enemy to be able to use the equipment to protect *their* secrets.

If you can put your algorithm in a tamperproof module, here's what you can do. You can require keys of a special and secret form; all other keys will cause the module to encrypt and decrypt using a severely weakened algorithm. You can make it so that the odds of someone, not knowing this special form but accidentally stumbling on a correct key, are vanishingly small.

This is called a **nonlinear keyspace**, because all the keys are not equally strong. (The opposite is a linear, or **flat**, keyspace.) An easy way to do this is to create the key as two parts: the key itself and some fixed string encrypted with that key. The module decrypts the string with the key; if it gets the fixed string it uses the key normally, if not it uses a different, weak algorithm. If the algorithm has a 128-bit key and a 64-bit block size, the overall key is 192 bits; this gives the algorithm an effective key of 2^{128}, but makes the odds of randomly choosing a good key one in 2^{64}.

You can be even subtler. You can design an algorithm such that certain keys are stronger than others. An algorithm can have no weak keys—keys that are obviously very poor—and can still have a nonlinear keyspace.

This only works if the algorithm is secret and the enemy can't reverse-engineer it, or if the difference in key strength is subtle enough that the enemy can't figure it out. The NSA did this with the secret algorithms in their Overtake modules (see Section 25.1). Did they do the same thing with Skipjack (see Section 13.12)? No one knows.

8.3 TRANSFERRING KEYS

Alice and Bob are going to use a symmetric cryptographic algorithm to communicate securely; they need the same key. Alice generates a key using a random-key generator. Now she has to give it to Bob—securely. If Alice can meet Bob somewhere (a back alley, a windowless room, or one of Jupiter's moons), she can give him a copy of the key. Otherwise, they have a problem. Public-key cryptography solves the problem nicely and with a minimum of prearrangement, but these techniques are not always available (see Section 3.1). Some systems use alternate channels known to be secure. Alice could send Bob the key with a trusted messenger. She could send it by certified mail or via an overnight delivery service. She could set up another communications channel with Bob and hope no one is eavesdropping on that one.

Alice could send Bob the symmetric key over their communications channel—the one they are going to encrypt. This is foolish; if the channel warrants encryption, sending the encryption key in the clear over the same channel guarantees that anyone eavesdropping on the channel can decrypt all communications.

The X9.17 standard [55] specifies two types of keys: key-encryption keys and data keys. **Key-Encryption Keys** encrypt other keys for distribution. **Data Keys** encrypt message traffic. These key-encrypting keys have to be distributed manually (although they can be secured in a tamperproof device, like a smart card), but only seldomly. Data keys are distributed more often. More details are in [75]. This two-tiered key concept is used a lot in key distribution.

Another solution to the distribution problem splits the key into several different parts (see Section 3.6) and sends each of those parts over a different channel. One part could be sent over the telephone, one by mail, one by overnight delivery service, one by carrier pigeon, and so on. (see Figure 8.2). Since an adversary could collect all but one of the parts and still have no idea what the key is, this method will work in all but extreme cases. Section 3.6 discusses schemes for splitting a key into several parts. Alice could even use a secret sharing scheme (see Section 3.7), allowing Bob to reconstruct the key if some of the shares are lost in transmission.

Alice sends Bob the key-encryption key securely, either by a face-to-face meeting or the splitting technique just discussed. Once Alice and Bob both have the key-encryption key, Alice can send Bob daily data keys over the same communications channel. Alice encrypts each data key with the key-encryption key. Since the amount of traffic being encrypted with the key-encryption key is low, it does not have to be changed as often. However, since compromise of the key-encryption key could compromise every message encrypted with every key that was encrypted with the key-encryption key, it must be stored securely.

Key Distribution in Large Networks

Key-encryption keys shared by pairs of users work well in small networks, but can quickly get cumbersome if the networks become large. Since every pair of users must exchange keys, the total number of key exchanges required in an n-person network is $n(n-1)/2$.

In a six-person network, 15 key exchanges are required. In a 1000-person network, nearly 500,000 key exchanges are required. In these cases, creating a central key server (or servers) makes the operation much more efficient.

Alternatively, any of the symmetric-cryptography or public-key-cryptography protocols in Section 3.1 provides for secure key distribution.

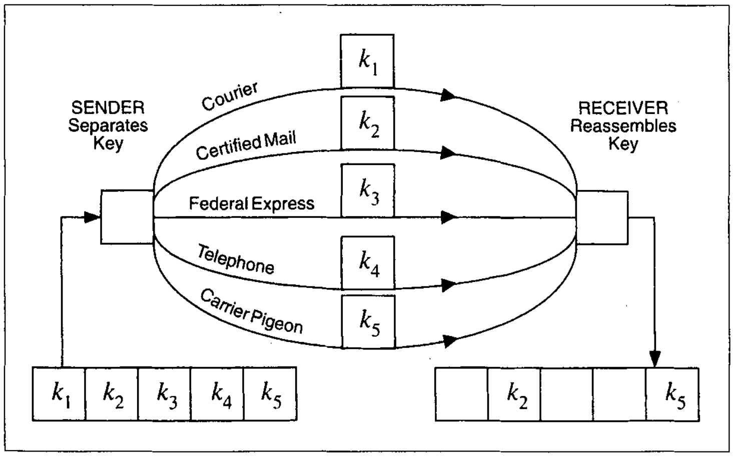


Figure 8.2 Key distribution via parallel channels.

8.4 VERIFYING KEYS

When Bob receives a key, how does he know it came from Alice and not from someone pretending to be Alice? If Alice gives it to him when they are face-to-face, it's easy. If Alice sends her key via a trusted courier, then Bob has to trust the courier. If the key is encrypted with a key-encryption key, then Bob has to trust the fact that only Alice has that key. If Alice uses a digital signature protocol to sign the key, Bob has to trust the public-key database when he verifies that signature. (He also has to trust that Alice has kept her key secure.) If a Key Distribution Center (KDC) signs Alice's public key, Bob has to trust that his copy of the KDC's public key has not been tampered with.

In the end, someone who controls the entire network around Bob can make him think whatever he likes. Mallory could send an encrypted and signed message purporting to be from Alice. When Bob tried to access the public-key database to verify Alice's signature, Mallory could substitute his own public key. Mallory could invent his own false KDC and exchange the real KDC's public key for his own creation. Bob wouldn't be the wiser.

Some people have used this argument to claim that public-key cryptography is useless. Since the only way for Alice and Bob to ensure that their keys have not been tampered with is to meet face-to-face, public-key cryptography doesn't enhance security at all.

This view is naïve. It is theoretically true, but reality is far more complicated. Public-key cryptography, used with digital signatures and trusted KDCs, makes it much more difficult to substitute one key for another. Bob can never be absolutely certain that Mallory isn't controlling his entire reality, but Bob can be confident that doing so requires more resources than most real-world Mallorys have access to.

Bob could also verify Alice's key over the telephone, where he can hear her voice. Voice recognition is a really good authentication scheme. If it's a public key, he can safely recite it in public. If it's a secret key, he can use a one-way hash function to verify the key. Both PGP (see Section 24.12) and the AT&T TSD (see Section 24.18) use this kind of key verification.

Sometimes, it may not even be important to verify exactly whom a public key belongs to. It may be necessary to verify that it belongs to the same person to whom it belonged last year. If someone sends a signed withdrawal message to a bank, the bank does not have to be concerned with who withdraws the money, only whether it is the same person who deposited the money in the first place.

Error Detection during Key Transmission

Sometimes keys get garbled in transmission. Since a garbled key can mean megabytes of undecryptable ciphertext, this is a problem. All keys should be transmitted with some kind of error detection and correction bits. This way errors in transmission can be easily detected and, if required, the key can be resent.

One of the most widely used methods is to encrypt a constant value with the key, and to send the first 2 to 4 bytes of that ciphertext along with the key. At the receiving end, do the same thing. If the encrypted constants match, then the key has been transmitted without error. The chance of an undetected error ranges from one in 2^{16} to one in 2^{32}.

Key-error Detection during Decryption

Sometimes the receiver wants to check if a particular key he has is the correct symmetric decryption key. If the plaintext message is something like ASCII, he can try to decrypt and read the message. If the plaintext is random, there are other tricks.

The naïve approach is to attach a **verification block**: a known header to the plaintext message before encryption. At the receiving end, Bob decrypts the header and verifies that it is correct. This works, but it gives Eve a known plaintext to help cryptanalyze the system. It also makes attacks against short-key ciphers like DES and all exportable ciphers easy. Precalculate the checksum once for each key, then use that checksum to determine the key in any message you intercept after that. This is a feature of *any* key checksum that doesn't include random or at least different data in each checksum. It's very similar in concept to using salt when generating keys from passphrases.

Here's a better way to do this [821]:

(1) Generate an IV (not the one used for the message).
(2) Use that IV to generate a large block of bits: say, 512.
(3) Hash the result.
(4) Use the same fixed bits of the hash, say 32, for the key checksum.

This gives Eve some information, but very little. If she tries to use the low 32 bits of the final hash value to mount a brute-force attack, she has to do multiple encryptions plus a hash per candidate key; brute-force on the key itself would be quicker.

She also gets no known-plaintext values to try out, and even if she manages to choose our random value for us, she never gets a chosen-plaintext out of us, since it goes through the hash function before she sees it.

8.5 USING KEYS

Software encryption is scary. Gone are the days of simple microcomputers under the control of single programs. Now there's Macintosh System 7, Windows NT, and UNIX. You can't tell when the operating system will suspend the encryption application in progress, write everything to disk, and take care of some pressing task. When the operating system finally gets back to encrypting whatever is being encrypted, everything will look just fine. No one will ever realize that the operating system wrote the encryption application to disk, and that it wrote the key along with it. The key will sit on the disk, unencrypted, until the computer writes over that area of memory again. It could be minutes or it could be months. It could even be never; the key could still be sitting there when an adversary goes over the hard drive with a fine-tooth comb. In a preemptive, multitasking environment, you can set your encryption operation to a high enough priority so it will not be interrupted. This would mitigate the risk. Even so, the whole thing is dicey at best.

Hardware implementations are safer. Many encryption devices are designed to erase the key if tampered with. For example, the IBM PS/2 encryption card has an epoxy unit containing the DES chip, battery, and memory. Of course, you have to trust the hardware manufacturer to implement the feature properly.

Some communications applications, such as telephone encryptors, can use **session keys**. A session key is a key that is just used for one communications session—a single telephone conversation—and then discarded. There is no reason to store the key after it has been used. And if you use some key-exchange protocol to transfer the key from one conversant to the other, the key doesn't have to be stored before it is used either. This makes it far less likely that the key might be compromised.

Controlling Key Usage

In some applications it may be desirable to control how a session key is used. Some users may need session keys only for encryption or only for decryption. Session keys might only be authorized for use on a certain machine or at a certain time. One scheme to handle these sorts of restrictions attaches a **Control Vector** (CV) to the key; the control vector specifies the uses and restrictions for that key (see Section 24.1) [1025,1026]. This CV is hashed and XORed with a master key; the result is used as an encryption key to encrypt the session key. The resultant encrypted session key is then stored with the CV. To recover the session key, hash the CV and XOR it with the master key, and use the result to decrypt the encrypted session key.

The advantages of this scheme are that the CV can be of arbitrary length and that it is always stored in the clear with the encrypted key. This scheme assumes quite a bit about tamperproof hardware and the inability of users to get at the keys directly. This system is discussed further in Sections 24.1 and 24.8.

8.6 UPDATING KEYS

Imagine an encrypted data link where you want to change keys daily. Sometimes it's a pain to distribute a new key every day. An easier solution is to generate a new key from the old key; this is sometimes called **key updating**.

All it takes is a one-way function. If Alice and Bob share the same key and they both operate on it using the same one-way function, they will get the same result. Then they can take the bits they need from the results to create the new key.

Key updating works, but remember that the new key is only as secure as the old key was. If Eve managed to get her hands on the old key, she can perform the key updating function herself. However, if Eve doesn't have the old key and is trying a ciphertext-only attack on the encrypted traffic, this is a good way for Alice and Bob to protect themselves.

8.7 STORING KEYS

The least complex key storage problem is that of a single user, Alice, encrypting files for later use. Since she is the only person involved, she is the only person responsible for the key. Some systems take the easy approach: The key is stored in Alice's brain and never on the system. Alice is responsible for remembering the key and entering it every time she needs a file encrypted or decrypted.

An example of this system is IPS [881]. Users can either directly enter the 64-bit key or enter the key as a longer character string. The system then generates a 64-bit key from the character string using a key-crunching technique.

Another solution is to store the key in a magnetic stripe card, plastic key with an embedded ROM chip (called a **ROM key**), or smart card [556,557,455]. A user could then enter his key into the system by inserting the physical token into a special reader in his encryption box or attached to his computer terminal. While the user can use the key, he does not know it and cannot compromise it. He can use it only in the way and for the purposes indicated by the control vector.

A ROM key is a very clever idea. People understand physical keys, what they signify and how to protect them. Putting a cryptographic key in the same physical form makes storing and protecting that key more intuitive.

This technique is made more secure by splitting the key into two halves, storing one half in the terminal and the other half in the ROM key. The U.S. government's STU-III secure telephone works this way. Losing the ROM key does not compromise the cryptographic key—change that key and everything is back to normal. The same is true with the loss of the terminal. This way, compromising either the ROM key or the system does not compromise the cryptographic key—an adversary must have both parts.

Hard-to-remember keys can be stored in encrypted form, using something similar to a key-encryption key. For example, an RSA private key could be encrypted with a DES key and stored on disk. To recover the RSA key, the user has to type in the DES key to a decryption program.

If the keys are generated deterministically (with a cryptographically secure pseudo-random-sequence generator), it might be easier to regenerate the keys from an easy-to-remember password every time they are required.

Ideally, a key should never appear unencrypted outside the encryption device. This isn't always possible, but it is a worthy goal.

8.8 BACKUP KEYS

Alice is the chief financial officer at Secrets, Ltd.—"We don't tell you our motto." Like any good corporate officer, she follows the company's security guidelines and encrypts all her data. Unfortunately, she ignores the company's street-crossing guidelines and gets hit by a truck. What does the company's president, Bob, do?

Unless Alice left a copy of her key, he's in deep trouble. The whole point of encryption is to make files unrecoverable without the key. Unless Alice was a moron and used lousy encryption software, her files are gone forever.

Bob can avoid this in several ways. The simplest is sometimes called **key escrow** (see Section 4.14): He requires all employees to write their keys on paper and give them to the company's security officer, who will lock them in a safe somewhere (or encrypt them all with a master key). Now, when Alice is bowled over on the Interstate, Bob can ask his security officer for her key. Bob should make sure to have the combination to the safe himself as well; otherwise, if the security officer is run over by another truck, Bob will be out of luck again.

The problem with this key management system is that Bob has to trust his security officer not to misuse everyone's keys. Even more significantly, all the employees have to trust the security officer not to misuse their keys. A far better solution is to use a secret-sharing protocol (see Section 3.7).

When Alice generates a key, she also divides up that key into some number of pieces. She then sends each piece—encrypted, of course—to a different company officer. None of those pieces alone is the key, but someone can gather all the pieces together and reconstruct the key. Now Alice is protected against any one malicious person, and Bob is protected against losing all of Alice's data after her run-in with the truck. Or, she could just store the different pieces, encrypted with each of the officer's different public keys, on her own hard disk. That way, no one gets involved with key management until it becomes necessary.

Another backup scheme [188] uses smart cards (see Section 24.13) for the temporary escrow of keys. Alice can put the key to secure her hard drive onto the smart card and give it to Bob while she is away. Bob can use the card to get into Alice's hard drive, but because the key is stored in the card Bob cannot learn it. And the system is bilaterally auditable: Bob can verify that the key will open Alice's drive, and when Alice returns she can verify if Bob has used the key and how many times.

Such a scheme makes no sense for data transmission. On a secure telephone, the key should exist for the length of the call and no longer. For data storage, as just described, key escrow can be a good idea. I've lost about one key every five years, and my memory is better than most. If 200 million people were using cryptography, that same rate would equal 40 million lost keys per year. I keep copies of my house keys with a neighbor because I may lose mine. If house keys were like cryptographic keys, and I lost them, I could never get inside and recover my possessions, ever again. Just as I keep off-site backups of my data, it makes sense to keep backups of my data-encryption keys.

8.9 COMPROMISED KEYS

All of the protocols, techniques, and algorithms in this book are secure only if the key (the private key in a public-key system) remains secret. If Alice's key is lost, stolen, printed in the newspaper, or otherwise compromised, then all her security is gone.

If the compromised key was for a symmetric cryptosystem, Alice has to change her key and hope the actual damage was minimal. If it was a private key, she has bigger problems; her public key is probably on servers all over the network. And if Eve gets access to Alice's private key, she can impersonate her on the network: reading encrypted mail, signing correspondence, entering into contracts, and so forth. Eve can, effectively, become Alice.

It is vital that news of a private key's compromise propagate quickly throughout the network. Any databases of public keys must immediately be notified that a particular private key has been compromised, lest some unsuspecting person encrypt a message in that compromised key.

One hopes Alice knows when her key was compromised. If a KDC is managing the keys, Alice should notify it that her key has been compromised. If there is no KDC, then she should notify all correspondents who might receive messages from her. Someone should publicize the fact that any message received after her key was lost is suspect, and that no one should send messages to Alice with the associated public key. The application should be using some sort of timestamp, and then users can determine which messages are legitimate and which are suspect.

If Alice doesn't know exactly when her key was compromised, things are more difficult. Alice may want to back out of a contract because the person who stole the key signed it instead of her. If the system allows this, then anyone can back out of a contract by claiming that his key was compromised before it was signed. It has to be a matter for an adjudicator to decide.

This is a serious problem and brings to light the dangers of Alice tying all of her identity to a single key. It would be better for Alice to have different keys for different applications—just as she has different physical keys in her pocket for different locks. Other solutions to this problem involve biometrics, limits on what can be done with a key, time delays, and countersigning.

These procedures and tips are hardly optimal, but are the best we can do. The moral of the story is to protect keys, and protect private keys above all else.

8.10 LIFETIME OF KEYS

No encryption key should be used for an indefinite period. It should expire automatically like passports and licenses. There are several reasons for this:

— The longer a key is used, the greater the chance that it will be compromised. People write keys down; people lose them. Accidents happen. If you use the same key for a year, there's a far greater chance of compromise than if you use it for a day.

— The longer a key is used, the greater the loss if the key is compromised. If a key is used only to encrypt a single budgetary document on a file server, then the loss of the key means only the compromise of that document. If the same key is used to encrypt all the budgetary information on the file server, then its loss is much more devastating.

— The longer a key is used, the greater the temptation for someone to spend the effort necessary to break it—even if that effort is a brute-force attack. Breaking a key shared between two military units for a day would enable someone to read and fabricate messages between those units for that day. Breaking a key shared by an entire military command structure for a year would enable that same person to read and fabricate messages throughout the world for a year. In our budget-conscious, post-Cold War world, which key would you choose to attack?

— It is generally easier to do cryptanalysis with more ciphertext encrypted with the same key.

For any cryptographic application, there must be a policy that determines the permitted lifetime of a key. Different keys may have different lifetimes. For a connection-based system, like a telephone, it makes sense to use a key for the length of the telephone call and to use a new one with each call.

Systems on dedicated communications channels are not as obvious. Keys should have relatively short lifetimes, depending on the value of the data and the amount of data encrypted during a given period. The key for a gigabit-per-second communications link might have to be changed more often than the key for a 9600-baud modem link. Assuming there is an efficient method of transferring new keys, session keys should be changed at least daily.

Key-encryption keys don't have to be replaced as frequently. They are used only occasionally (roughly once per day) for key exchange. This generates little ciphertext for a cryptanalyst to work with, and the corresponding plaintext has no particular form. However, if a key-encryption key is compromised, the potential loss is extreme: all communications encrypted with every key encrypted with the key-encryption key. In some applications, key-encryption keys are replaced only once a month or once a year. You have to balance the inherent danger in keeping a key around for a while with the inherent danger in distributing a new one.

Encryption keys used to encrypt data files for storage cannot be changed often. The files may sit encrypted on disk for months or years before someone needs them again. Decrypting them and re-encrypting them with a new key every day doesn't enhance security in any way; it just gives a cryptanalyst more to work with. One solution might be to encrypt each file with a unique file key, and then encrypt all the file keys with a key-encryption key. The key-encryption key should then be either memorized or stored in a secure location, perhaps in a safe somewhere. Of course, losing this key would mean losing all the individual file keys.

Private keys for public-key cryptography applications have varying lifetimes, depending on the application. Private keys used for digital signatures and proofs of identity may have to last years (even a lifetime). Private keys used for coin-flipping protocols can be discarded immediately after the protocol is completed. Even if a key's security is expected to last a lifetime, it may be prudent to change the key every couple of years. The private keys in many networks are good only for two years; after that the user must get a new private key. The old key would still have to remain secret, in case the user needed to verify a signature from that period. But the new key would be used to sign new documents, reducing the number of signed documents a cryptanalyst would have for an attack.

8.11 DESTROYING KEYS

Given that keys must be replaced regularly, old keys must be destroyed. Old keys are valuable, even if they are never used again. With them, an adversary can read old messages encrypted with those keys [65].

Keys must be destroyed securely (see Section 10.9). If the key is written on paper, the paper should be shredded or burned. Be careful to use a high-quality shredder; many lousy shredders are on the market. Algorithms in this book are secure against brute-force attacks costing millions of dollars and taking millions of years. If an adversary can recover your key by taking a bag of shredded documents from your trash and paying 100 unemployed workers in some backwater country ten cents per hour for a year to piece the shredded pages together, that would be $26,000 well spent.

If the key is in a hardware EEPROM, the key should be overwritten multiple times. If the key is in a hardware EPROM or PROM, the chip should be smashed into tiny bits and scattered to the four winds. If the key is stored on a computer disk, the actual bits of the storage should be overwritten multiple times (see Section 10.9) or the disk should be shredded.

A potential problem is that, in a computer, keys can be easily copied and stored in multiple locations. Any computer that does its own memory management, constantly swapping programs in and out of memory, exacerbates the problem. There is no way to ensure that successful key erasure has taken place in the computer, especially if the computer's operating system controls the erasure process. The more paranoid among you should consider writing a special erasure program that scans all disks looking for copies of the key's bit pattern on unused blocks and then erases those blocks. Also remember to erase the contents of any temporary, or "swap," files.

8.12 PUBLIC-KEY KEY MANAGEMENT

Public-key cryptography makes key management easier, but it has its own unique problems. Each person has only one public key, regardless of the number of people on the network. If Alice wants to send a message to Bob, she has to get Bob's public key. She can go about this several ways:

- She can get it from Bob.
- She can get it from a centralized database.
- She can get it from her own private database.

Section 2.5 discussed a number of possible attacks against public-key cryptography, based on Mallory substituting his key for Bob's. The scenario is that Alice wants to send a message to Bob. She goes to the public-key database and gets Bob's public key. But Mallory, who is sneaky, has substituted his own key for Bob's. (If Alice asks Bob directly, Mallory has to intercept Bob's transmission and substitute his key for Bob's.) Alice encrypts her message in Mallory's key and sends it to Bob. Mallory intercepts the message, decrypts it, and reads it. He re-encrypts it with Bob's real key and sends it on to Bob. Neither Alice nor Bob is the wiser.

Public-key Certificates

A **public-key certificate** is someone's public key, signed by a trustworthy person. Certificates are used to thwart attempts to substitute one key for another [879]. Bob's certificate, in the public-key database, contains a lot more than his public key. It contains information about Bob—his name, address, and so on—and it is signed by someone Alice trusts: Trent (usually known as a **certification authority**, or CA). By signing both the key and the information about Bob, Trent certifies that the information about Bob is correct and that the public key belongs to Bob. Alice checks Trent's signature and then uses the public key, secure in the knowledge that it is Bob's and no one else's. Certificates play an important role in a number of public-key protocols such as PEM [825] (see Section 24.10) and X.509 [304] (see Section 24.9).

A complicated noncryptographic issue surrounds this type of system. What is the meaning of certification? Or, to put it another way, who is trusted to issue certificates to whom? Anyone may sign anyone else's certificate, but there needs to be some way to filter out questionable certificates: for example, certificates for employees of one company signed by the CA of another company. Normally, a certification chain transfers trust: A single trusted entity certifies trusted agents, trusted agents certify company CAs, and company CAs certify their employees.

Here are some more things to think about:

- What level of trust in someone's identity is implied by his certificate?
- What are the relationships between a person and the CA that certified his public key, and how can those relationships be implied by the certificate?
- Who can be trusted to be the "single trusted entity" at the top of the certification chain?
- How long can a certification chain be?

Ideally, Bob would follow some kind of authentication procedure before the CA signs his certificate. Additionally, some kind of timestamp or an indication of the certificate's validity period is important to guard against compromised keys [461].

Timestamping is not enough. Keys may be invalidated before they have expired, either through compromise or for administrative reasons. Hence, it is important the CA keep a list of invalid certificates, and for users to regularly check that list. This key revocation problem is still a difficult one to solve.

And one public-key/private-key pair is not enough. Certainly any good implementation of public-key cryptography needs separate keys for encryption and digital signatures. This separation allows for different security levels, expiration times, backup procedures, and so on. Someone might sign messages with a 2048-bit key stored on a smart card and good for twenty years, while they might use a 768-bit key stored in the computer and good for six months for encryption.

And a single pair of encryption and signature keys isn't enough, either. A private key authenticates a relationship as well as an identity, and people have more than one relationship. Alice might want to sign one document as Alice the individual, another as Alice, vice-president of Monolith, Inc., and a third as Alice, president of her community organization. Some of these keys are more valuable than others, so they can be better protected. Alice might have to store a backup of her work key with the company's security officer; she doesn't want the company to have a copy of the key she signed her mortgage with. Just as Alice has multiple physical keys in her pocket, she is going to have multiple cryptographic keys.

Distributed Key Management

In some situations, this sort of centralized key management will not work. Perhaps there is no CA whom Alice and Bob both trust. Perhaps Alice and Bob trust only their friends. Perhaps Alice and Bob trust no one.

Distributed key management, used in PGP (see Section 24.12), solves this problem with **introducers**. Introducers are other users of the system who sign their friends' public keys. For example, when Bob generates his public key, he gives copies to his friends: Carol and Dave. They know Bob, so they each sign Bob's key and give Bob a copy of the signature. Now, when Bob presents his key to a stranger, Alice, he presents it with the signatures of these two introducers. If Alice also knows and trusts Carol, she has reason to believe that Bob's key is valid. If she knows and trusts Carol and Dave a little, she has reason to believe that Bob's key is valid. If she doesn't know either Carol or Dave, she has no reason to trust Bob's key.

Over time, Bob will collect many more introducers. If Alice and Bob travel in similar circles, the odds are good that Alice will know one of Bob's introducers. To prevent against Mallory's substituting one key for another, an introducer must be sure that Bob's key belongs to Bob before he signs it. Perhaps the introducer should require the key be given face-to-face or verified over the telephone.

The benefit of this mechanism is that there is no CA that everyone has to trust. The down side is that when Alice receives Bob's public key, she has no guarantee that she will know any of the introducers and therefore no guarantee that she will trust the validity of the key.

References

1. ABA Bank Card Standard, "Management and Use of Personal Information Numbers," Aids from ABA, Catalog no. 207213, American Bankers Association, 1979.
2. ABA Document 4.3, "Key Management Standard," American Bankers Association, 1980.
3. M. Abadi, J. Feigenbaum, and J. Kilian, "On Hiding Information from an Oracle," *Proceedings of the 19th ACM Symposium on the Theory of Computing*, 1987, pp. 195–203.
4. M. Abadi, J. Feigenbaum, and J. Kilian, "On Hiding Information from an Oracle," *Journal of Computer and System Sciences*, v. 39, n. 1, Aug 1989, pp. 21–50.
5. M. Abadi and R. Needham, "Prudent Engineering Practice for Cryptographic Protocols," Research Report 125, Digital Equipment Corp Systems Research Center, Jun 1994.
6. C.M. Adams, "On Immunity Against Biham and Shamir's 'Differential Cryptanalysis,' " *Information Processing Letters*, v. 41, 14 Feb 1992, pp. 77–80.
7. C.M. Adams, "Simple and Effective Key Scheduling for Symmetric Ciphers," *Workshop on Selected Areas in Cryptography—Workshop Record*, Kingston, Ontario, 5–6 May 1994, pp. 129–133.
8. C.M. Adams and H. Meijer, "Security-Related Comments Regarding McEliece's Public-Key Cryptosystem," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 224–230.
9. C.M. Adams and S.E. Tavares, "The Structured Design of Cryptographically Good S-Boxes," *Journal of Cryptology*, v. 3, n. 1, 1990, pp. 27–41.
10. C.M. Adams and S.E. Tavares, "Designing S-Boxes for Ciphers Resistant to Differential Cryptanalysis," *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography*, Rome, Italy, 15–16 Feb 1993, pp. 181–190.
11. W. Adams and D. Shanks, "Strong Primality Tests That Are Not Sufficient," *Mathematics of Computation*, v. 39, 1982, pp. 255–300.
12. W.W. Adams and L.J. Goldstein, *Introduction to Number Theory*, Englewood Cliffs, N.J.: Prentice-Hall, 1976.
13. B.S. Adiga and P. Shankar, "Modified Lu-Lee Cryptosystem," *Electronics Letters*, v. 21, n. 18, 29 Aug 1985, pp. 794–795.
14. L.M. Adleman, "A Subexponential Algorithm for the Discrete Logarithm Problem with Applications to Cryptography," *Proceedings of the IEEE 20th Annual Symposium of Foundations of Computer Science*, 1979, pp. 55–60.
15. L.M. Adleman, "On Breaking Generalized Knapsack Public Key Cryptosystems," *Proceedings of the 15th ACM Symposium on Theory of Computing*, 1983, pp. 402–412.

16. L.M. Adleman, "Factoring Numbers Using Singular Integers," *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing*, 1991, pp. 64–71.

17. L.M. Adleman, "Molecular Computation of Solutions to Combinatorial Problems," *Science*, v. 266, n. 11, Nov 1994, p. 1021.

18. L.M. Adleman, D. Estes, and K. McCurley, "Solving Bivariate Quadratic Congruences in Random Polynomial Time," *Mathematics of Computation*, v. 48, n. 177, Jan 1987, pp. 17–28.

19. L.M. Adleman, C. Pomerance, and R.S. Rumeley, "On Distinguishing Prime Numbers from Composite Numbers," *Annals of Mathematics*, v. 117, n. 1, 1983, pp. 173–206.

20. L.M. Adleman and R.L. Rivest, "How to Break the Lu-Lee (COMSAT) Public-Key Cryptosystem," MIT Laboratory for Computer Science, Jul 1979.

21. G.B. Agnew, "Random Sources for Cryptographic Systems," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 77–81.

22. G.B. Agnew, R.C. Mullin, I.M. Onyszchuk, and S.A. Vanstone, "An Implementation for a Fast Public-Key Cryptosystem," *Journal of Cryptology*, v. 3, n. 2, 1991, pp. 63–79.

23. G.B. Agnew, R.C. Mullin, and S.A. Vanstone, "A Fast Elliptic Curve Cryptosystem," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 706–708.

24. G.B. Agnew, R.C. Mullin, and S.A. Vanstone, "Improved Digital Signature Scheme Based on Discrete Exponentiation," *Electronics Letters*, v. 26, n. 14, 5 Jul 1990, pp. 1024–1025.

25. G.B. Agnew, R.C. Mullin, and S.A. Vanstone, "On the Development of a Fast Elliptic Curve Cryptosystem," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 482–287.

26. G.B. Agnew, R.C. Mullin, and S.A. Vanstone, "An Implementation of Elliptic Curve Cryptosystems over F_2155," *IEEE Selected Areas of Communications*, v. 11, n. 5, Jun 1993, pp. 804–813.

27. A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

28. S.G. Akl, "Digital Signatures: A Tutorial Survey," *Computer*, v. 16, n. 2, Feb 1983, pp. 15–24.

29. S.G. Akl, "On the Security of Compressed Encodings," *Advances in Cryptology: Proceedings of Crypto 83*, Plenum Press, 1984, pp. 209–230.

30. S.G. Akl and H. Meijer, "A Fast Pseudo-Random Permutation Generator with Applications to Cryptology," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 269–275.

31. M. Alabbadi and S.B. Wicker, "Security of Xinmei Digital Signature Scheme," *Electronics Letters*, v. 28, n. 9, 23 Apr 1992, pp. 890–891.

32. M. Alabbadi and S.B. Wicker, "Digital Signature Schemes Based on Error-Correcting Codes," *Proceedings of the 1993 IEEE-ISIT*, IEEE Press, 1993, p. 199.

33. M. Alabbadi and S.B. Wicker, "Cryptanalysis of the Harn and Wang Modification of the Xinmei Digital Signature Scheme," *Electronics Letters*, v. 28, n. 18, 27 Aug 1992, pp. 1756–1758.

34. K. Alagappan and J. Tardo, "SPX Guide: Prototype Public Key Authentication Service," Digital Equipment Corp., May 1991.

35. W. Alexi, B.-Z. Chor, O. Goldreich, and C.P. Schnorr, "RSA and Rabin Functions: Certain Parts Are as Hard as the Whole," *Proceedings of the 25th IEEE Symposium on the Foundations of Computer Science*, 1984, pp. 449–457.

36. W. Alexi, B.-Z. Chor, O. Goldreich, and C.P. Schnorr, "RSA and Rabin Functions: Certain Parts are as Hard as the Whole," *SIAM Journal on Computing*, v. 17, n. 2, Apr 1988, pp. 194–209.

37. Ameritech Mobile Communications et al., "Cellular Digital Packet Data System Specifications: Part 406: Airlink Security," CDPD Industry Input Coordinator, Costa Mesa, Calif., Jul 1993.

38. H.R. Amirazizi, E.D. Karnin, and J.M. Reyneri, "Compact Knapsacks are Polynomial Solvable," *ACM SIGACT News*, v. 15, 1983, pp. 20–22.

39. R.J. Anderson, "Solving a Class of Stream Ciphers," *Cryptologia*, v. 14, n. 3, Jul 1990, pp. 285–288.

40. R.J. Anderson, "A Second Generation Electronic Wallet," *ESORICS 92, Proceedings of the Second European Symposium on*

Research in Computer Security, Springer-Verlag, 1992, pp. 411–418.

41. R.J. Anderson, "Faster Attack on Certain Stream Ciphers," *Electronics Letters*, v. 29, n. 15, 22 Jul 1993, pp. 1322–1323.
42. R.J. Anderson, "Derived Sequence Attacks on Stream Ciphers," presented at the rump session of CRYPTO '93, Aug 1993.
43. R.J. Anderson, "Why Cryptosystems Fail," *1st ACM Conference on Computer and Communications Security*, ACM Press, 1993, pp. 215–227.
44. R.J. Anderson, "Why Cryptosystems Fail," *Communications of the ACM*, v. 37, n. 11, Nov 1994, pp. 32–40.
45. R.J. Anderson, "On Fibonacci Keystream Generators," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
46. R.J. Anderson, "Searching for the Optimum Correlation Attack," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
47. R.J. Anderson and T.M.A. Lomas, "Fortifying Key Negotiation Schemes with Poorly Chosen Passwords," *Electronics Letters*, v. 30, n. 13, 23 Jun 1994, pp. 1040–1041.
48. R.J. Anderson and R. Needham, "Robustness Principles for Public Key Protocols," *Advances in Cryptology—CRYPTO '95 Proceedings*, Springer-Verlag, 1995, to appear.
49. D. Andleman and J. Reeds, "On the Cryptanalysis of Rotor Machines and Substitution-Permutation Networks," *IEEE Transactions on Information Theory*, v. IT-28, n. 4, Jul 1982, pp. 578–584.
50. ANSI X3.92, "American National Standard for Data Encryption Algorithm (DEA)," American National Standards Institute, 1981.
51. ANSI X3.105, "American National Standard for Information Systems—Data Link Encryption," American National Standards Institute, 1983.
52. ANSI X3.106, "American National Standard for Information Systems—Data Encryption Algorithm—Modes of Operation," American National Standards Institute, 1983.
53. ANSI X9.8, "American National Standard for Personal Information Number (PIN) Management and Security," American Bankers Association, 1982.
54. ANSI X9.9 (Revised), "American National Standard for Financial Institution Message Authentication (Wholesale)," American Bankers Association, 1986.
55. ANSI X9.17 (Revised), "American National Standard for Financial Institution Key Management (Wholesale)," American Bankers Association, 1985.
56. ANSI X9.19, "American National Standard for Retail Message Authentication," American Bankers Association, 1985.
57. ANSI X9.23, "American National Standard for Financial Institution Message Encryption," American Bankers Association, 1988.
58. ANSI X9.24, "Draft Proposed American National Standard for Retail Key Management," American Bankers Association, 1988.
59. ANSI X9.26 (Revised), "American National Standard for Financial Institution Sign-On Authentication for Wholesale Financial Transaction," American Bankers Association, 1990.
60. ANSI X9.30, "Working Draft: Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry," American Bankers Association, Aug 1994.
61. ANSI X9.31, "Working Draft: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry," American Bankers Association, Mar 1993.
62. K. Aoki and K. Ohta, "Differential-Linear Cryptanalysis of FEAL-8," *Proceedings of the 1995 Symposium on Cryptography and Information Security (SCIS 95)*, Inuyama, Japan, 24–27 Jan 1995, pp. A3.4.1-11. (In Japanese.)
63. K. Araki and T. Sekine, "On the Conspiracy Problem of the Generalized Tanaka's Cryptosystem," *IEICE Transactions*, v. E74, n. 8, Aug 1991, pp. 2176–2178.
64. S. Araki, K. Aoki, and K. Ohta, "The Best Linear Expression Search for FEAL," *Proceedings of the 1995 Symposium on Cryptography and Information Security (SCIS 95)*, Inuyama, Japan, 24–27 Jan 1995, pp. A4.4.1-10.
65. C. Asmuth and J. Bloom, "A Modular Approach to Key Safeguarding," *IEEE Transactions on Information Theory*, v. IT-29, n. 2, Mar 1983, pp. 208–210.

66. D. Atkins, M. Graff, A.K. Lenstra, and P.C. Leyland, "The Magic Words are Squeamish Ossifrage," *Advances in Cryptology—ASIACRYPT '94 Proceedings*, Springer-Verlag, 1995, pp. 263–277.
67. AT&T, "T7001 Random Number Generator," Data Sheet, Aug 1986.
68. AT&T, "AT&T Readying New Spy-Proof Phone for Big Military and Civilian Markets," *The Report on AT&T*, 2 Jun 1986, pp. 6–7.
69. AT&T, "T7002/T7003 Bit Slice Multiplier," product announcement, 1987.
70. AT&T, "Telephone Security Device TSD 3600—User's Manual," AT&T, 20 Sep 1992.
71. Y. Aumann and U. Feige, "On Message Proof Systems with Known Space Verifiers," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 85–99.
72. R.G. Ayoub, *An Introduction to the Theory of Numbers*, Providence, RI: American Mathematical Society, 1963.
73. A. Aziz and W. Diffie, "Privacy and Authentication for Wireless Local Area Networks," *IEEE Personal Communications*, v. 1, n. 1, 1994, pp. 25–31.
74. A. Bahreman and J.D. Tygar, "Certified Electronic Mail," *Proceedings of the Internet Society 1994 Workshop on Network and Distributed System Security*, The Internet Society, 1994, pp. 3–19.
75. D. Balenson, "Automated Distribution of Cryptographic Keys Using the Financial Institution Key Management Standard," *IEEE Communications Magazine*, v. 23, n. 9, Sep 1985, pp. 41–46.
76. D. Balenson, "Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers," RFC 1423, Feb 1993.
77. D. Balenson, C.M. Ellison, S.B. Lipner, and S.T. Walker, "A New Approach to Software Key Escrow Encryption," TIS Report #520, Trusted Information Systems, Aug 94.
78. R. Ball, *Mathematical Recreations and Essays*, New York: MacMillan, 1960.
79. J. Bamford, *The Puzzle Palace*, Boston: Houghton Mifflin, 1982.
80. J. Bamford and W. Madsen, *The Puzzle Palace*, Second Edition, Penguin Books, 1995.
81. S.K. Banerjee, "High Speed Implementation of DES," *Computers & Security*, v. 1, 1982, pp. 261–267.
82. Z. Baodong, "MC-Veiled Linear Transform Public Key Cryptosystem," *Acta Electronica Sinica*, v. 20, n. 4, Apr 1992, pp. 21–24. (In Chinese.)
83. P.H. Bardell, "Analysis of Cellular Automata Used as Pseudorandom Pattern Generators," *Proceedings of 1990 International Test Conference*, pp. 762–768.
84. T. Baritaud, H. Gilbert, and M. Girault, "FFT Hashing is not Collision-Free," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 35–44.
85. C. Barker, "An Industry Perspective of the CCEP," *2nd Annual AIAA Computer Security Conference Proceedings*, 1986.
86. W.G. Barker, *Cryptanalysis of the Hagelin Cryptograph*, Aegean Park Press, 1977.
87. P. Barrett, "Implementing the Rivest Shamir and Adleman Public Key Encryption Algorithm on a Standard Digital Signal Processor," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 311–323.
88. T.C. Bartee and D.I. Schneider, "Computation with Finite Fields," *Information and Control*, v. 6, n. 2, Jun 1963, pp. 79–98.
89. U. Baum and S. Blackburn, "Clock-Controlled Pseudorandom Generators on Finite Groups," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
90. K.R. Bauer, T.A. Bersen, and R.J. Feiertag, "A Key Distribution Protocol Using Event Markers," *ACM Transactions on Computer Systems*, v. 1, n. 3, 1983, pp. 249–255.
91. F. Bauspiess and F. Damm, "Requirements for Cryptographic Hash Functions," *Computers & Security*, v. 11, n. 5, Sep 1992, pp. 427–437.
92. D. Bayer, S. Haber, and W.S. Stornetta, "Improving the Efficiency and Reliability of Digital Time-Stamping," *Sequences '91: Methods in Communication, Security, and Computer Science*, Springer-Verlag, 1992, pp. 329–334.
93. R. Bayer and J.K. Metzger, "On the Encipherment of Search Trees and Random Access Files," *ACM Transactions on Database Systems*, v. 1, n. 1, Mar 1976, pp. 37–52.

94. M. Beale and M.F. Monaghan, "Encrytion Using Random Boolean Functions," *Cryptography and Coding*, H.J. Beker and F.C. Piper, eds., Oxford: Clarendon Press, 1989, pp. 219–230.
95. P. Beauchemin and G. Brassard, "A Generalization of Hellman's Extension to Shannon's Approach to Cryptography," *Journal of Cryptology*, v. 1, n. 2, 1988, pp. 129–132.
96. P. Beauchemin, G. Brassard, C. Crépeau, C. Goutier, and C. Pomerance, "The Generation of Random Numbers that are Probably Prime," *Journal of Cryptology*, v. 1, n. 1, 1988, pp. 53–64.
97. D. Beaver, J. Feigenbaum, and V. Shoup, "Hiding Instances in Zero-Knowledge Proofs," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 326–338.
98. H. Beker, J. Friend, and P. Halliden, "Simplifying Key Management in Electronic Funds Transfer Points of Sale Systems," *Electronics Letters*, v. 19, n. 12, Jun 1983, pp. 442–444.
99. H. Beker and F. Piper, *Cipher Systems: The Protection of Communications*, London: Northwood Books, 1982.
100. D.E. Bell and L.J. LaPadula, "Secure Computer Systems: Mathematical Foundations," Report ESD-TR-73-275, MITRE Corp., 1973.
101. D.E. Bell and L.J. LaPadula, "Secure Computer Systems: A Mathematical Model," Report MTR-2547, MITRE Corp., 1973.
102. D.E. Bell and L.J. LaPadula, "Secure Computer Systems: A Refinement of the Mathematical Model," Report ESD-TR-73-278, MITRE Corp., 1974.
103. D.E. Bell and L.J. LaPadula, "Secure Computer Systems: Unified Exposition and Multics Interpretation," Report ESD-TR-75-306, MITRE Corp., 1976.
104. M. Bellare and S. Goldwasser, "New Paradigms for Digital Signatures and Message Authentication Based on Non-Interactive Zero Knowledge Proofs," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 194–211.
105. M. Bellare and S. Micali, "Non-Interactive Oblivious Transfer and Applications," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 547–557.
106. M. Bellare, S. Micali, and R. Ostrovsky, "Perfect Zero-Knowledge in Constant Rounds," *Proceedings of the 22nd ACM Symposium on the Theory of Computing*, 1990, pp. 482–493.
107. S.M. Bellovin, "A Preliminary Technical Analysis of Clipper and Skipjack," unpublished manuscript, 20 Apr 1993.
108. S.M. Bellovin and M. Merritt, "Limitations of the Kerberos Protocol," *Winter 1991 USENIX Conference Proceedings*, USENIX Association, 1991, pp. 253–267.
109. S.M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," *Proceedings of the 1992 IEEE Computer Society Conference on Research in Security and Privacy*, 1992, pp. 72–84.
110. S.M. Bellovin and M. Merritt, "An Attack on the Interlock Protocol When Used for Authentication," *IEEE Transactions on Information Theory*, v. 40, n. 1, Jan 1994, pp. 273–275.
111. S.M. Bellovin and M. Merritt, "Cryptographic Protocol for Secure Communications," U.S. Patent #5,241,599, 31 Aug 93.
112. I. Ben-Aroya and E. Biham, "Differential Cryptanalysis of Lucifer," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 187–199.
113. J.C. Benaloh, "Cryptographic Capsules: A Disjunctive Primitive for Interactive Protocols," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, 213–222.
114. J.C. Benaloh, "Secret Sharing Homorphisms: Keeping Shares of a Secret Secret," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 251–260.
115. J.C. Benaloh, "Verifiable Secret-Ballot Elections," Ph.D. dissertation, Yale University, YALEU/DCS/TR-561, Dec 1987.
116. J.C. Benaloh and M. de Mare, "One-Way Accumulators: A Decentralized Alternative to Digital Signatures," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 274–285.
117. J.C. Benaloh and D. Tuinstra, "Receipt-Free Secret Ballot Elections," *Proceedings of the 26th ACM Symposium on the Theory of Computing*, 1994, pp. 544–553.
118. J.C. Benaloh and M. Yung, "Distributing the Power of a Government to Enhance

the Privacy of Voters," *Proceedings of the 5th ACM Symposium on the Principles in Distributed Computing,* 1986, pp. 52–62.

119. A. Bender and G. Castagnoli, "On the Implementation of Elliptic Curve Cryptosystems," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 186–192.
120. S. Bengio, G. Brassard, Y.G. Desmedt, C. Goutier, and J.-J. Quisquater, "Secure Implementation of Identification Systems," *Journal of Cryptology,* v. 4, n. 3, 1991, pp. 175–184.
121. C.H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental Quantum Cryptography," *Advances in Cryptology—EUROCRYPT '90 Proceedings,* Springer-Verlag, 1991, pp. 253–265.
122. C.H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental Quantum Cryptography," *Journal of Cryptology,* v. 5, n. 1, 1992, pp. 3–28.
123. C.H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing,* Banjalore, India, Dec 1984, pp. 175–179.
124. C.H. Bennett and G. Brassard, "An Update on Quantum Cryptography," *Advances in Cryptology: Proceedings of CRYPTO 84,* Springer-Verlag, 1985, pp. 475–480.
125. C.H. Bennett and G. Brassard, "Quantum Public-Key Distribution System," *IBM Technical Disclosure Bulletin,* v. 28, 1985, pp. 3153–3163.
126. C.H. Bennett and G. Brassard, "Quantum Public Key Distribution Reinvented," *SIGACT News,* v. 18, n. 4, 1987, pp. 51–53.
127. C.H. Bennett and G. Brassard, "The Dawn of a New Era for Quantum Cryptography: The Experimental Prototype is Working!" *SIGACT News,* v. 20, n. 4, Fall 1989, pp. 78–82.
128. C.H. Bennett, G. Brassard, and S. Breidbart, *Quantum Cryptography II: How to Re-Use a One-Time Pad Safely Even if* ***P=NP***, unpublished manuscript, Nov 1982.
129. C.H. Bennett, G. Brassard, S. Breidbart, and S. Weisner, "Quantum Cryptography, or Unforgeable Subway Tokens," *Advances in Cryptology: Proceedings of Crypto 82,* Plenum Press, 1983, pp. 267–275.
130. C.H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska, "Practical Quantum Oblivious Transfer," *Advances in Cryptology—CRYPTO '91 Proceedings,* Springer-Verlag, 1992, pp. 351–366.
131. C.H. Bennett, G. Brassard, and A.K. Ekert, "Quantum Cryptography," *Scientific American,* v. 267, n. 4, Oct 1992, pp. 50–57.
132. C.H. Bennett, G. Brassard, and N.D. Mermin, "Quantum Cryptography Without Bell's Theorem," *Physical Review Letters,* v. 68, n. 5, 3 Feb 1992, pp. 557–559.
133. C.H. Bennett, G. Brassard, and J.-M. Robert, "How to Reduce Your Enemy's Information," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 468–476.
134. C.H. Bennett, G. Brassard, and J.-M. Robert, "Privacy Amplification by Public Discussion," *SIAM Journal on Computing,* v. 17, n. 2, Apr 1988, pp. 210–229.
135. J. Bennett, "Analysis of the Encryption Algorithm Used in WordPerfect Word Processing Program," *Cryptologia,* v. 11, n. 4, Oct 1987, pp. 206–210.
136. M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation," *Proceedings of the 20th ACM Symposium on the Theory of Computing,* 1988, pp. 1–10.
137. M. Ben-Or, O. Goldreich, S. Goldwasser, J. Håstad, J. Kilian, S. Micali, and P. Rogaway, "Everything Provable is Provable in Zero-Knowledge," *Advances in Cryptology—CRYPTO '88 Proceedings,* Springer-Verlag, 1990, pp. 37–56.
138. M. Ben-Or, O. Goldreich, S. Micali, and R.L. Rivest, "A Fair Protocol for Signing Contracts," *IEEE Transactions on Information Theory,* v. 36, n. 1, Jan 1990, pp. 40–46.
139. H.A. Bergen and W.J. Caelli, "File Security in WordPerfect 5.0," *Cryptologia,* v. 15, n. 1, Jan 1991, pp. 57–66.
140. E.R. Berlekamp, *Algebraic Coding Theory,* Aegean Park Press, 1984.
141. S. Berkovits, "How to Broadcast a Secret," *Advances in Cryptology—EUROCRYPT '91 Proceedings,* Springer-Verlag, 1991, pp. 535–541.
142. S. Berkovits, J. Kowalchuk, and B. Schanning, "Implementing Public-Key Scheme," *IEEE Communications Magazine,* v. 17, n. 3, May 1979, pp. 2–3.

143. D.J. Bernstein, Bernstein vs. U.S. Department of State et al., Civil Action No. C95-0582-MHP, United States District Court for the Northern District of California, 21 Feb 1995.
144. T. Berson, "Differential Cryptanalysis Mod 2^{32} with Applications to MD5," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, 1992, pp. 71–80.
145. T. Beth, *Verfahren der schnellen Fourier-Transformation*, Teubner, Stuttgart, 1984. (In German.)
146. T. Beth, "Efficient Zero-Knowledge Identification Scheme for Smart Cards," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 77–84.
147. T. Beth, B.M. Cook, and D. Gollmann, "Architectures for Exponentiation in GF(2^n)," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 302–310.
148. T. Beth and Y. Desmedt, "Identification Tokens—or: Solving the Chess Grandmaster Problem," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 169–176.
149. T. Beth and C. Ding, "On Almost Nonlinear Permutations," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 65–76.
150. T. Beth, M. Frisch, and G.J. Simmons, eds., *Lecture Notes in Computer Science 578; Public Key Cryptography: State of the Art and Future Directions*, Springer-Verlag, 1992.
151. T. Beth and F.C. Piper, "The Stop-and-Go Generator," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1984, pp. 88–92.
152. T. Beth and F. Schaefer, "Non Supersingular Elliptic Curves for Public Key Cryptosystems," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 316–327.
153. A. Beutelspacher, "How to Say 'No'," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 491–496.
154. J. Bidzos, letter to NIST regarding DSS, 20 Sep 1991.
155. J. Bidzos, personal communication, 1993.
156. P. Bieber, "A Logic of Communication in a Hostile Environment," *Proceedings of the Computer Security Foundations Workshop III*, IEEE Computer Society Press, 1990, pp. 14–22.
157. E. Biham, "Cryptanalysis of the Chaotic-Map Cryptosystem Suggested at EUROCRYPT '91," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 532–534.
158. E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys," Technical Report #753, Computer Science Department, Technion—Israel Institute of Technology, Sep 1992.
159. E. Biham, "On the Applicability of Differential Cryptanalysis to Hash Functions," lecture at EIES Workshop on Cryptographic Hash Functions, Mar 1992.
160. E. Biham, personal communication, 1993.
161. E. Biham, "Higher Order Differential Cryptanalysis," unpublished manuscript, Jan 1994.
162. E. Biham, "On Modes of Operation," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 116–120.
163. E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys," *Journal of Cryptology*, v. 7, n. 4, 1994, pp. 229–246.
164. E. Biham, "On Matsui's Linear Cryptanalysis," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, pp. 398–412.
165. E. Biham and A. Biryukov, "How to Strengthen DES Using Existing Hardware," *Advances in Cryptology—ASIACRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.
166. E. Biham and P.C. Kocher, "A Known Plaintext Attack on the PKZIP Encryption," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
167. E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 2–21.
168. E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," *Journal of Cryptology*, v. 4, n. 1, 1991, pp 3–72.
169. E. Biham and A. Shamir, "Differential Cryptanalysis of Feal and N-Hash," *Advances in Cryptology—EUROCRYPT*

'91 Proceedings, Springer-Verlag, 1991, pp. 1–16.

170. E. Biham and A. Shamir, "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI, and Lucifer," *Advances in Cryptology—CRYPTO '91 Proceedings*, 1992, pp. 156–171.
171. E. Biham and A. Shamir, "Differential Cryptanalysis of the Full 16-Round DES," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, 487–496.
172. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
173. R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, "Systematic Design of Two-Party Authentication Protocols," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 44–61.
174. R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, "Systematic Design of a Family of Attack-Resistant Authentication Protocols," *IEEE Journal of Selected Areas in Communication*, to appear.
175. R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, "A Modular Family of Secure Protocols for Authentication and Key Distribution," *IEEE/ACM Transactions on Networking*, to appear.
176. M. Bishop, "An Application for a Fast Data Encryption Standard Implementation," *Computing Systems*, v. 1, n. 3, 1988, pp. 221–254.
177. M. Bishop, "Privacy-Enhanced Electronic Mail," *Distributed Computing and Cryptography*, J. Feigenbaum and M. Merritt, eds., American Mathematical Society, 1991, pp. 93–106.
178. M. Bishop, "Privacy-Enhanced Electronic Mail," *Internetworking: Research and Experience*, v. 2, n. 4, Dec 1991, pp. 199–233.
179. M. Bishop, "Recent Changes to Privacy Enhanced Electronic Mail," *Internetworking: Research and Experience*, v. 4, n. 1, Mar 1993, pp. 47–59.
180. I.F. Blake, R. Fuji-Hara, R.C. Mullin, and S.A. Vanstone, "Computing Logarithms in Finite Fields of Characteristic Two," *SIAM Journal on Algebraic Discrete Methods*, v. 5, 1984, pp. 276–285.
181. I.F. Blake, R.C. Mullin, and S.A. Vanstone, "Computing Logarithms in GF (2^n)," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 73–82.
182. G.R. Blakley, "Safeguarding Cryptographic Keys," *Proceedings of the National Computer Conference, 1979*, American Federation of Information Processing Societies, v. 48, 1979, pp. 313–317.
183. G.R. Blakley, "One-Time Pads are Key Safeguarding Schemes, Not Cryptosystems—Fast Key Safeguarding Schemes (Threshold Schemes) Exist," *Proceedings of the 1980 Symposium on Security and Privacy*, IEEE Computer Society, Apr 1980, pp. 108–113.
184. G.R. Blakley and I. Borosh, "Rivest-Shamir-Adleman Public Key Cryptosystems Do Not Always Conceal Messages," *Computers and Mathematics with Applications*, v. 5, n. 3, 1979, pp. 169–178.
185. G.R. Blakley and C. Meadows, "A Database Encryption Scheme which Allows the Computation of Statistics Using Encrypted Data," *Proceedings of the 1985 Symposium on Security and Privacy*, IEEE Computer Society, Apr 1985, pp. 116–122.
186. M. Blaze, "A Cryptographic File System for UNIX," *1st ACM Conference on Computer and Communications Security*, ACM Press, 1993, pp. 9–16.
187. M. Blaze, "Protocol Failure in the Escrowed Encryption Standard," *2nd ACM Conference on Computer and Communications Security*, ACM Press, 1994, pp. 59–67.
188. M. Blaze, "Key Management in an Encrypting File System," *Proceedings of the Summer 94 USENIX Conference*, USENIX Association, 1994, pp. 27–35.
189. M. Blaze and B. Schneier, "The MacGuffin Block Cipher Algorithm," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
190. U. Blöcher and M. Dichtl, "Fish: A Fast Software Stream Cipher," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 41–44.
191. R. Blom, "Non-Public Key Distribution," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 231–236.

192. K.J. Blow and S.J.D. Phoenix, "On a Fundamental Theorem of Quantum Cryptography," *Journal of Modern Optics*, v. 40, n. 1, Jan 1993, pp. 33–36.
193. L. Blum, M. Blum, and M. Shub, "A Simple Unpredictable Pseudo-Random Number Generator," *SIAM Journal on Computing*, v. 15, n. 2, 1986, pp. 364–383.
194. M. Blum, "Coin Flipping by Telephone: A Protocol for Solving Impossible Problems," *Proceedings of the 24th IEEE Computer Conference (CompCon)*, 1982, pp. 133–137.
195. M. Blum, "How to Exchange (Secret) Keys," *ACM Transactions on Computer Systems*, v. 1, n. 2, May 1983, pp. 175–193.
196. M. Blum, "How to Prove a Theorem So No One Else Can Claim It," *Proceedings of the International Congress of Mathematicians*, Berkeley, CA, 1986, pp. 1444–1451.
197. M. Blum, A. De Santis, S. Micali, and G. Persiano, "Noninteractive Zero-Knowledge," *SIAM Journal on Computing*, v. 20, n. 6, Dec 1991, pp. 1084–1118.
198. M. Blum, P. Feldman, and S. Micali, "Non-Interactive Zero-Knowledge and Its Applications," *Proceedings of the 20th ACM Symposium on Theory of Computing*, 1988, pp. 103–112.
199. M. Blum and S. Goldwasser, "An *Efficient* Probabilistic Public-Key Encryption Scheme Which Hides All Partial Information," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 289–299.
200. M. Blum and S. Micali, "How to Generate Cryptographically-Strong Sequences of Pseudo-Random Bits," *SIAM Journal on Computing*, v. 13, n. 4, Nov 1984, pp. 850–864.
201. B. den Boer, "Cryptanalysis of F.E.A.L.," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 293–300.
202. B. den Boer and A. Bosselaers, "An Attack on the Last Two Rounds of MD4," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 194–203.
203. B. den Boer and A. Bosselaers, "Collisions for the Compression Function of MD5," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 293–304.
204. J.-P. Boly, A. Bosselaers, R. Cramer, R. Michelsen, S. Mjølsnes, F. Muller, T. Pedersen, B. Pfitzmann, P. de Rooij, B. Schoenmakers, M. Schunter, L. Vallée, and M. Waidner, "Digital Payment Systems in the ESPRIT Project CAFE," *Securicom 94*, Paris, France, 2–6 Jan 1994, pp. 35–45.
205. J.-P. Boly, A. Bosselaers, R. Cramer, R. Michelsen, S. Mjølsnes, F. Muller, T. Pedersen, B. Pfitzmann, P. de Rooij, B. Schoenmakers, M. Schunter, L. Vallée, and M. Waidner, "The ESPRIT Project CAFE—High Security Digital Payment System," *Computer Security—ESORICS 94*, Springer-Verlag, 1994, pp. 217–230.
206. D.J. Bond, "Practical Primality Testing," *Proceedings of IEE International Conference on Secure Communications Systems*, 22–23 Feb 1984, pp. 50–53.
207. H. Bonnenberg, *Secure Testing of VSLI Cryptographic Equipment*, Series in Microelectronics, Vol. 25, Konstanz: Hartung Gorre Verlag, 1993.
208. H. Bonnenberg, A. Curiger, N. Felber, H. Kaeslin, and X. Lai, "VLSI Implementation of a New Block Cipher," *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD 91)*, Oct 1991, pp. 510–513.
209. K.S. Booth, "Authentication of Signatures Using Public Key Encryption," *Communications of the ACM*, v. 24, n. 11, Nov 1981, pp. 772–774.
210. A. Bosselaers, R. Govaerts, and J. Vandewalle, *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 175–186.
211. D.P. Bovet and P. Crescenzi, *Introduction to the Theory of Complexity*, Englewood Cliffs, N.J.: Prentice-Hall, 1994.
212. J. Boyar, "Inferring Sequences Produced by a Linear Congruential Generator Missing Low-Order Bits," *Journal of Cryptology*, v. 1, n. 3, 1989, pp. 177–184.
213. J. Boyar, D. Chaum, and I. Damgård, "Convertible Undeniable Signatures," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 189–205.
214. J. Boyar, K. Friedl, and C. Lund, "Practical Zero-Knowledge Proofs: Giving Hints and Using Deficiencies," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 155–172.

215. J. Boyar, C. Lund, and R. Peralta, "On the Communication Complexity of Zero-Knowledge Proofs," *Journal of Cryptology*, v. 6, n. 2, 1993, pp. 65–85.
216. J. Boyar and R. Peralta, "On the Concrete Complexity of Zero-Knowledge Proofs," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 507–525.
217. C. Boyd, "Some Applications of Multiple Key Ciphers," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 455–467.
218. C. Boyd, "Digital Multisignatures," *Cryptography and Coding*, H.J. Beker and F.C. Piper, eds., Oxford: Clarendon Press, 1989, pp. 241–246.
219. C. Boyd, "A New Multiple Key Cipher and an Improved Voting Scheme," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 617–625.
220. C. Boyd, "Multisignatures Revisited," *Cryptography and Coding III*, M.J. Ganley, ed., Oxford: Clarendon Press, 1993, pp. 21–30.
221. C. Boyd and W. Mao, "On the Limitation of BAN Logic," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 240–247.
222. C. Boyd and W. Mao, "Designing Secure Key Exchange Protocols," *Computer Security—ESORICS 94*, Springer-Verlag, 1994, pp. 217–230.
223. B.O. Brachtl, D. Coppersmith, M.M. Hyden, S.M. Matyas, C.H. Meyer, J. Oseas, S. Pilpel, and M. Schilling, "Data Authentication Using Modification Detection Codes Based on a Public One Way Function," U.S. Patent #4,908,861, 13 Mar 1990.
224. J. Brandt, I.B. Damgård, P. Landrock, and T. Pederson, "Zero-Knowledge Authentication Scheme with Secret Key Exchange," *Advances in Cryptology—CRYPTO '88*, Springer-Verlag, 1990, pp. 583–588.
225. S.A. Brands, "An Efficient Off-Line Electronic Cash System Based on the Representation Problem," Report CS-R9323, Computer Science/Department of Algorithms and Architecture, CWI, Mar 1993.
226. S.A. Brands, "Untraceable Off-line Cash in Wallet with Observers," *Advances in Cryptology—CRYPTO '93*, Springer-Verlag, 1994, pp. 302–318.
227. S.A. Brands, "Electronic Cash on the Internet," *Proceedings of the Internet Society 1995 Symposium on Network and Distributed Systems Security*, IEEE Computer Society Press 1995, pp 64–84.
228. D.K. Branstad, "Hellman's Data Does Not Support His Conclusion," *IEEE Spectrum*, v. 16, n. 7, Jul 1979, p. 39.
229. D.K. Branstad, J. Gait, and S. Katzke, "Report on the Workshop on Cryptography in Support of Computer Security," NBSIR 77-1291, National Bureau of Standards, Sep 21–22, 1976, September 1977.
230. G. Brassard, "A Note on the Complexity of Cryptography," *IEEE Transactions on Information Theory*, v. IT-25, n. 2, Mar 1979, pp. 232–233.
231. G. Brassard, "Relativized Cryptography," *Proceedings of the IEEE 20th Annual Symposium on the Foundations of Computer Science*, 1979, pp. 383–391.
232. G. Brassard, "A Time-Luck Tradeoff in Relativized Cryptography," *Proceedings of the IEEE 21st Annual Symposium on the Foundations of Computer Science*, 1980, pp. 380–386.
233. G. Brassard, "A Time-Luck Tradeoff in Relativized Cryptography," *Journal of Computer and System Sciences*, v. 22, n. 3, Jun 1981, pp. 280–311.
234. G. Brassard, "An Optimally Secure Relativized Cryptosystem," *SIGACT News*, v. 15, n. 1, 1983, pp. 28–33.
235. G. Brassard, "Relativized Cryptography," *IEEE Transactions on Information Theory*, v. IT-29, n. 6, Nov 1983, pp. 877–894.
236. G. Brassard, *Modern Cryptology: A Tutorial*, Springer-Verlag, 1988.
237. G. Brassard, "Quantum Cryptography: A Bibliography," *SIGACT News*, v. 24, n. 3, Oct 1993, pp. 16–20.
238. G. Brassard, D. Chaum, and C. Crépeau, "An Introduction to Minimum Disclosure," *CWI Quarterly*, v. 1, 1988, pp. 3–17.
239. G. Brassard, D. Chaum, and C. Crépeau, "Minimum Disclosure Proofs of Knowledge," *Journal of Computer and System Sciences*, v. 37, n. 2, Oct 1988, pp. 156–189.
240. G. Brassard and C. Crépeau, "Non-Transitive Transfer of Confidence: A *Perfect* Zero-Knowledge Interactive Protocol for SAT and Beyond," *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, 1986, pp. 188–195.

241. G. Brassard and C. Crépeau, "Zero-Knowledge Simulation of Boolean Circuits," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 223–233.
242. G. Brassard and C. Crépeau, "Sorting Out Zero-Knowledge," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 181–191.
243. G. Brassard and C. Crépeau, "Quantum Bit Commitment and Coin Tossing Protocols," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 49–61.
244. G. Brassard, C. Crépeau, R. Jozsa, and D. Langlois, "A Quantum Bit Commitment Scheme Provably Unbreakable by Both Parties," *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, 1993, pp. 362–371.
245. G. Brassard, C. Crépeau, and J.-M. Robert, "Information Theoretic Reductions Among Disclosure Problems," *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, 1986, pp. 168–173.
246. G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-Nothing Disclosure of Secrets," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 234–238.
247. G. Brassard, C. Crépeau, and M. Yung, "Everything in **NP** Can Be Argued in Perfect Zero-Knowledge in a Bounded Number of Rounds," *Proceedings on the 16th International Colloquium on Automata, Languages, and Programming*, Springer-Verlag, 1989, pp. 123–136.
248. R.P. Brent, "An Improved Monte-Carlo Factorization Algorithm," *BIT*, v. 20, n. 2, 1980, pp. 176–184.
249. R.P. Brent, "On the Periods of Generalized Fibonacci Recurrences, *Mathematics of Computation*, v. 63, n. 207, Jul 1994, pp. 389–401.
250. R.P. Brent, "Parallel Algorithms for Integer Factorization," Research Report CMA-R49-89, Computer Science Laboratory, The Australian National University, Oct 1989.
251. D.M. Bressoud, *Factorization and Primality Testing*, Springer-Verlag, 1989.
252. E.F. Brickell, "A Fast Modular Multiplication Algorithm with Applications to Two Key Cryptography," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1982, pp. 51–60.
253. E.F. Brickell, "Are Most Low Density Polynomial Knapsacks Solvable in Polynomial Time?" *Proceedings of the 14th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, 1983.
254. E.F. Brickell, "Solving Low Density Knapsacks," *Advances in Cryptology: Proceedings of Crypto 83*, Plenum Press, 1984, pp. 25–37.
255. E.F. Brickell, "Breaking Iterated Knapsacks," *Advances in Cryptology: Proceedings of Crypto 84*, Springer-Verlag, 1985, pp. 342–358.
256. E.F. Brickell, "Cryptanalysis of the Uagisawa Public Key Cryptosystem," *Abstracts of Papers, EUROCRYPT '86*, 20–22 May 1986.
257. E.F. Brickell, "The Cryptanalysis of Knapsack Cryptosystems," *Applications of Discrete Mathematics*, R.D. Ringeisen and F.S. Roberts, eds., Society for Industrial and Applied Mathematics, Philadelphia, 1988, pp. 3–23.
258. E.F. Brickell, "Survey of Hardware Implementations of RSA," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 368–370.
259. E.F. Brickell, D. Chaum, I.B. Damgård, and J. van de Graff, "Gradual and Verifiable Release of a Secret," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 156–166.
260. E.F. Brickell, J.A. Davis, and G.J. Simmons, "A Preliminary Report on the Cryptanalysis of Merkle-Hellman Knapsack," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 289–303.
261. E.F. Brickell and J. DeLaurentis, "An Attack on a Signature Scheme Proposed by Okamoto and Shiraishi," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 28–32.
262. E.F. Brickell, D.E. Denning, S.T. Kent, D.P. Maher, and W. Tuchman, "SKIPJACK Review—Interim Report," unpublished manuscript, 28 Jul 1993.
263. E.F. Brickell, J.C. Lagarias, and A.M. Odlyzko, "Evaluation of the Adleman Attack of Multiple Iterated Knapsack Cryptosystems," *Advances in Cryptology:*

Proceedings of Crypto 83, Plenum Press, 1984, pp. 39–42.

264. E.F. Brickell, P.J. Lee, and Y. Yacobi, "Secure Audio Teleconference," *Advances in Cryptology—CRYPTO '87 Proceedings,* Springer-Verlag, 1988, pp. 418–426.
265. E.F. Brickell and K.S. McCurley, "An Interactive Identification Scheme Based on Discrete Logarithms and Factoring," *Advances in Cryptology—EUROCRYPT '90 Proceedings,* Springer-Verlag, 1991, pp. 63–71.
266. E.F. Brickell, J.H. Moore, and M.R. Purtill, "Structure in the S-Boxes of the DES," *Advances in Cryptology—CRYPTO '86 Proceedings,* Springer-Verlag, 1987, pp. 3–8.
267. E.F. Brickell and A.M. Odlyzko, "Cryptanalysis: A Survey of Recent Results," *Proceedings of the IEEE,* v. 76, n. 5, May 1988, pp. 578–593.
268. E.F. Brickell and A.M. Odlyzko, "Cryptanalysis: A Survey of Recent Results," *Contemporary Cryptology: The Science of Information Integrity,* G.J. Simmons, ed., IEEE Press, 1991, pp. 501–540.
269. E.F. Brickell and G.J. Simmons, "A Status Report on Knapsack Based Public Key Cryptosystems," *Congressus Numerantium,* v. 7, 1983, pp. 3–72.
270. E.F. Brickell and D.R. Stinson, "The Detection of Cheaters in Threshold Schemes," *Advances in Cryptology—CRYPTO '88 Proceedings,* Springer-Verlag, 1990, pp. 564–577.
271. A.G. Broscius and J.M. Smith, "Exploiting Parallelism in Hardware Implementation of the DES," *Advances in Cryptology—CRYPTO '91 Proceedings,* Springer-Verlag, 1992, pp. 367–376.
272. L. Brown, M. Kwan, J. Pieprzyk, and J. Seberry, "Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI," *Advances in Cryptology—ASIACRYPT '91 Proceedings,* Springer-Verlag, 1993, pp. 36–50.
273. L. Brown, J. Pieprzyk, and J. Seberry, "LOKI: A Cryptographic Primitive for Authentication and Secrecy Applications," *Advances in Cryptology—AUSCRYPT '90 Proceedings,* Springer-Verlag, 1990, pp. 229–236.
274. L. Brown, J. Pieprzyk, and J. Seberry, "Key Scheduling in DES Type Cryptosystems," *Advances in Cryptology—AUSCRYPT '90 Proceedings,* Springer-Verlag, 1990, pp. 221–228.
275. L. Brown and J. Seberry, "On the Design of Permutation P in DES Type Cryptosystems," *Advances in Cryptology—EUROCRYPT '89 Proceedings,* Springer-Verlag, 1990, pp. 696–705.
276. W. Brown, "A Quantum Leap in Secret Communications," *New Scientist,* n. 1585, 30 Jan 1993, p. 21.
277. J.O. Brüer, "On Pseudo Random Sequences as Crypto Generators," *Proceedings of the International Zurich Seminar on Digital Communication,* Switzerland, 1984.
278. L. Brynielsson "On the Linear Complexity of Combined Shift Register Sequences," *Advances in Cryptology—EUROCRYPT '85,* Springer-Verlag, 1986, pp. 156–166.
279. J. Buchmann, J. Loho, and J. Zayer, "An Implementation of the General Number Field Sieve," *Advances in Cryptology—CRYPTO '93 Proceedings,* Springer-Verlag, 1994, pp. 159–165.
280. M. Burmester and Y. Desmedt, "Broadcast Interactive Proofs," *Advances in Cryptology—EUROCRYPT '91 Proceedings,* Springer-Verlag, 1991, pp. 81–95.
281. M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," *Advances in Cryptology—EUROCRYPT '94 Proceedings,* Springer-Verlag, 1995, to appear.
282. D. Burnham, "NSA Seeking 500,000 'Secure' Telephones," *The New York Times,* 6 Oct 1994.
283. M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," Research Report 39, Digital Equipment Corp. Systems Research Center, Feb 1989.
284. M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *ACM Transactions on Computer Systems,* v. 8, n. 1, Feb 1990, pp. 18–36.
285. M. Burrows, M. Abadi, and R. Needham, "Rejoinder to Nessett," *Operating System Review,* v. 20, n. 2, Apr 1990, pp. 39–40.
286. J.J. Cade, "A Modification of a Broken Public-Key Cipher," *Advances in Cryptology—CRYPTO '86 Proceedings,* Springer-Verlag, 1987, pp. 64–83.
287. T.R. Cain and A.T. Sherman, "How to Break Gifford's Cipher," *Proceedings of the 2nd Annual ACM Conference on*

Computer and Communications Security, ACM Press, 1994, pp. 198–209.

288. C. Calvelli and V. Varadharajan, "An Analysis of Some Delegation Protocols for Distributed Systems," *Proceedings of the Computer Security Foundations Workshop V,* IEEE Computer Society Press, 1992, pp. 92–110.
289. J.L. Camenisch, J.-M. Piveteau, and M.A. Stadler, "An Efficient Electronic Payment System Protecting Privacy," *Computer Security—ESORICS 94,* Springer-Verlag, 1994, pp. 207–215.
290. P. Camion and J. Patarin, "The Knapsack Hash Function Proposed at Crypto '89 Can Be Broken," *Advances in Cryptology—EUROCRYPT '91,* Springer-Verlag, 1991, pp. 39–53.
291. C.M. Campbell, "Design and Specification of Cryptographic Capabilities," *IEEE Computer Society Magazine,* v. 16, n. 6, Nov 1978, pp. 15–19.
292. E.A. Campbell, R. Safavi-Naini, and P.A. Pleasants, "Partial Belief and Probabilistic Reasoning in the Analysis of Secure Protocols," *Proceedings of the Computer Security Foundations Workshop V,* IEEE Computer Society Press, 1992, pp. 92–110.
293. K.W. Campbell and M.J. Wiener, "DES Is Not a Group," *Advances in Cryptology—CRYPTO '92 Proceedings,* Springer-Verlag, pp. 512–520.
294. Z.F. Cao and G. Zhao, "Some New MC Knapsack Cryptosystems," *CHINACRYPT '94,* Xidian, China, 11–15 Nov 1994, pp. 70–75. (In Chinese).
295. C. Carlet, "Partially-Bent Functions," *Advances in Cryptology—CRYPTO '92 Proceedings,* Springer-Verlag, 1993, pp. 280–291.
296. C. Carlet, "Partially Bent Functions," *Designs, Codes and Cryptography,* v. 3, 1993, pp. 135–145.
297. C. Carlet, "Two New Classes of Bent Functions" *Advances in Cryptology—EUROCRYPT '93 Proceedings,* Springer-Verlag, 1994, pp. 77–101.
298. C. Carlet, J. Seberry, and X.M. Zhang, "Comments on 'Generating and Counting Binary Bent Sequences,'" *IEEE Transactions on Information Theory,* v. IT-40, n. 2, Mar 1994, p. 600.
299. J.M. Carroll, *Computer Security,* 2nd edition, Butterworths, 1987.
300. J.M. Carroll, "The Three Faces of Information Security," *Advances in Cryptology—AUSCRYPT '90 Proceedings,* Springer-Verlag, 1990, pp. 433–450.
301. J.M. Carroll, "'Do-it-yourself' Cryptography," *Computers & Security,* v. 9, n. 7, Nov 1990, pp. 613–619.
302. T.R. Caron and R.D. Silverman, "Parallel Implementation of the Quadratic Scheme," *Journal of Supercomputing,* v. 1, n. 3, 1988, pp. 273–290.
303. CCITT, Draft Recommendation X.509, "The Directory—Authentication Framework," Consultation Committee, International Telephone and Telegraph, International Telecommunications Union, Geneva, 1987.
304. CCITT, Recommendation X.509, "The Directory—Authentication Framework," Consultation Committee, International Telephone and Telegraph, International Telecommunications Union, Geneva, 1989.
305. CCITT, Recommendation X.800, "Security Architecture for Open Systems Interconnection for CCITT Applications," International Telephone and Telegraph, International Telecommunications Union, Geneva, 1991.
306. F. Chabaud, "On the Security of Some Cryptosystems Based on Error-Correcting Codes," *Advances in Cryptology—EUROCRYPT '94 Proceedings,* Springer-Verlag, 1995, to appear.
307. F. Chabaud and S. Vaudenay, "Links Between Differential and Linear Cryptanalysis," *Advances in Cryptology—EUROCRYPT '94 Proceedings,* Springer-Verlag, 1995, to appear.
308. W.G. Chambers and D. Gollmann, "Generators for Sequences with Near-Maximal Linear Equivalence," *IEE Proceedings,* V. 135, Pt. E, n. 1, Jan 1988, pp. 67–69.
309. W.G. Chambers and D. Gollmann, "Lock-In Effect in Cascades of Clock-Controlled Shirt Registers," *Advances in Cryptology—EUROCRYPT '88 Proceedings,* Springer-Verlag, 1988, pp. 331–343.
310. A. Chan and R. Games, "On the Linear Span of Binary Sequences from Finite Geometries," *Advances in Cryptology—CRYPTO '86 Proceedings,* Springer-Verlag, 1987, pp. 405–417.
311. J.P. Chandler, D.C. Arrington, D.R. Berkelhammer, and W.L. Gill, "Identification and

Analysis of Foreign Laws and Regulations Pertaining to the Use of Commercial Encryption Products for Voice and Data Communications," National Intellectual Property Law Institute, George Washington University, Washington, D.C., Jan 1994.

312. C.C. Chang and S.J. Hwang, "Cryptographic Authentication of Passwords," *Proceedings of the 25th Annual 1991 IEEE International Carnahan Conference on Security Technology*, Taipei, Taiwan, 1–3 Oct 1991, pp. 126–130.

313. C.C. Chang and S.J. Hwang, "A Strategy for Transforming Public-Key Cryptosystems into Identity-Based Cryptosystems," *Proceedings of the 25th Annual 1991 IEEE International Carnahan Conference on Security Technology*, Taipei, Taiwan, 1–3 Oct 1991, pp. 68–72.

314. C.C. Chang and C.H. Lin, "An ID-Based Signature Scheme Based upon Rabin's Public Key Cryptosystem," *Proceedings of the 25th Annual 1991 IEEE International Carnahan Conference on Security Technology*, Taipei, Taiwan, 1–3 Oct 1991, pp. 139–141.

315. C. Charnes and J. Pieprzyk, "Attacking the SL_2 Hashing Scheme," *Advances in Cryptology—ASIACRYPT '94 Proceedings*, Springer-Verlag, 1995, pp. 322–330.

316. D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, v. 24, n. 2, Feb 1981, pp. 84–88.

317. D. Chaum, "Blind Signatures for Untraceable Payments," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 199–203.

318. D. Chaum, "Security Without Identification: Transaction Systems to Make Big Brother Obsolete," *Communications of the ACM*, v. 28, n. 10, Oct 1985, pp. 1030–1044.

319. D. Chaum, "Demonstrating that a Public Predicate Can Be Satisfied without Revealing Any Information about How," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 159–199.

320. D. Chaum, "Blinding for Unanticipated Signatures," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 227–233.

321. D. Chaum, "The Dining Cryptographers Problem: Unconditional Sender and Receiver Untraceability," *Journal of Cryptology*, v. 1, n. 1, 1988, pp. 65–75.

322. D. Chaum, "Elections with Unconditionally Secret Ballots and Disruptions Equivalent to Breaking RSA," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 177–181.

323. D. Chaum, "Blind Signature Systems," U.S. Patent #4,759,063, 19 Jul 1988.

324. D. Chaum, "Blind Unanticipated Signature Systems," U.S. Patent #4,759,064, 19 Jul 1988.

325. D. Chaum, "Online Cash Checks," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 288–293.

326. D. Chaum, "One-Show Blind Signature Systems," U.S. Patent #4,914,698, 3 Apr 1990.

327. D. Chaum, "Undeniable Signature Systems," U.S. Patent #4,947,430, 7 Aug 1990.

328. D. Chaum, "Returned-Value Blind Signature Systems," U.S. Patent #4,949,380, 14 Aug 1990.

329. D. Chaum, "Zero-Knowledge Undeniable Signatures," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 458–464.

330. D. Chaum, "Group Signatures," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 257–265.

331. D. Chaum, "Unpredictable Blind Signature Systems," U.S. Patent #4,991,210, 5 Feb 1991.

332. D. Chaum, "Achieving Electronic Privacy," *Scientific American*, v. 267, n. 2, Aug 1992, pp. 96–101.

333. D. Chaum, "Designated Confirmer Signatures," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.

334. D. Chaum, C. Crépeau, and I.B. Damgård, "Multiparty Unconditionally Secure Protocols," Proceedings of the 20th ACM Symposium on the Theory of Computing, 1988, pp. 11–19.

335. D. Chaum, B. den Boer, E. van Heyst, S. Mjølsnes, and A. Steenbeek, "Efficient Offline Electronic Checks," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 294–301.

336. D. Chaum and J.-H. Evertse, "Cryptanalysis of DES with a Reduced Number of Rounds; Sequences of Linear Factors in Block Ciphers," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 192–211.
337. D. Chaum, J.-H. Evertse, and J. van de Graff, "An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 127–141.
338. D. Chaum, J.-H. Evertse, J. van de Graff, and R. Peralta, "Demonstrating Possession of a Discrete Logarithm without Revealing It," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 200–212.
339. D. Chaum, A. Fiat, and M. Naor, "Untraceable Electronic Cash," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 319–327.
340. D. Chaum and T. Pedersen, "Transferred Cash Grows in Size," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 391–407.
341. D. Chaum and T. Pedersen, "Wallet Databases with Observers," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, pp. 89–105.
342. D. Chaum and I. Schaumuller-Bichel, eds., *Smart Card 2000*, North Holland: Elsevier Science Publishers, 1989.
343. D. Chaum and H. van Antwerpen, "Undeniable Signatures," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 212–216.
344. D. Chaum, E. van Heijst, and B. Pfitzmann, "Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 470–484.
345. T.M. Chee, "The Cryptanalysis of a New Public-Key Cryptosystem Based on Modular Knapsacks," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 204–212.
346. L. Chen, "Oblivious Signatures," *Computer Security—ESORICS 94*, Springer-Verlag, 1994, pp. 161–172.
347. L. Chen and M. Burminster, "A Practical Secret Voting Scheme which Allows Voters to Abstain," *CHINACRYPT '94*, Xidian, China, 11–15 Nov 1994, pp. 100–107.
348. L. Chen and T.P. Pedersen "New Group Signature Schemes," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.
349. J. Chenhui, "Spectral Characteristics of Partially-Bent Functions," *CHINACRYPT '94*, Xidian, China, 11–15 Nov 1994, pp. 48–51.
350. V. Chepyzhov and B. Smeets, "On a Fast Correlation Attack on Certain Stream Ciphers," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 176–185.
351. T.C. Cheung, "Management of PEM Public Key Certificates Using X.500 Directory Service: Some Problems and Solutions," *Proceedings of the Internet Society 1994 Workshop on Network and Distributed System Security*, The Internet Society, 1994, pp. 35–42.
352. G.C. Chiou and W.C. Chen, "Secure Broadcasting Using the Secure Lock," *IEEE Transactions on Software Engineering*, v. SE-15, n. 8, Aug 1989, pp. 929–934.
353. Y.J. Choie and H.S. Hwoang, "On the Cryptosystem Using Elliptic Curves," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 105–113.
354. B. Chor and O. Goldreich, "RSA/Rabin Least Significant Bits are 1/2+1/poly(log *N*) Secure," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 303–313.
355. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults," *Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science*, 1985, pp. 383–395.
356. B. Chor and R.L. Rivest, "A Knapsack Type Public Key Cryptosystem Based on Arithmetic in Finite Fields," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 54–65.
357. P. Christoffersson, S.-A. Ekahll, V. Fåk, S. Herda, P. Mattila, W. Price, and H.-O. Widman, *Crypto Users' Handbook: A Guide for Implementors of Cryptographic Protection in Computer Systems*, North Holland: Elsevier Science Publishers, 1988.

358. R. Cleve, "Controlled Gradual Disclosure Schemes for Random Bits and Their Applications," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 572–588.

359. J.D. Cohen, "Improving Privacy in Cryptographic Elections," Yale University Computer Science Department Technical Report YALEU/DCS/TR-454, Feb 1986.

360. J.D. Cohen and M.H. Fischer, "A Robust and Verifiable Cryptographically Secure Election Scheme," *Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science*, 1985, pp. 372–382.

361. R. Cole, "A Model for Security in Distributed Systems," *Computers and Security*, v. 9, n. 4, Apr 1990, pp. 319–330.

362. Comptroller General of the United States, "Matter of National Institute of Standards and Technology—Use of Electronic Data Interchange Technology to Create Valid Obligations," File B-245714, 13 Dec 1991.

363. M.S. Conn, letter to Joe Abernathy, National Security Agency, Ser: Q43-111-92, 10 Jun 1992.

364. C. Connell, "An Analysis of NewDES: A Modified Version of DES," *Cryptologia*, v. 14, n. 3, Jul 1990, pp. 217–223.

365. S.A. Cook, "The Complexity of Theorem-Proving Procedures," *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, 1971, pp. 151–158.

366. R.H. Cooper and W. Patterson, "A Generalization of the Knapsack Method Using Galois Fields," *Cryptologia*, v. 8, n. 4, Oct 1984, pp. 343–347.

367. R.H. Cooper and W. Patterson, "RSA as a Benchmark for Multiprocessor Machines," *Advances in Cryptology—AUSCRYPT '90 Proceedings*, Springer-Verlag, 1990, pp. 356–359.

368. D. Coppersmith, "Fast Evaluation of Logarithms in Fields of Characteristic Two," *IEEE Transactions on Information Theory*, v. 30, n. 4, Jul 1984, pp. 587–594.

369. D. Coppersmith, "Another Birthday Attack," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 14–17.

370. D. Coppersmith, "Cheating at Mental Poker," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 104–107.

371. D. Coppersmith, "The Real Reason for Rivest's Phenomenon," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 535–536.

372. D. Coppersmith, "Two Broken Hash Functions," Research Report RD 18397, IBM T.J. Watson Center, Oct 1992.

373. D. Coppersmith, "The Data Encryption Standard (DES) and Its Strength against Attacks," Technical Report RC 18613, IBM T.J. Watson Center, Dec 1992.

374. D. Coppersmith, "The Data Encryption Standard (DES) and its Strength against Attacks," *IBM Journal of Research and Development*, v. 38, n. 3, May 1994, pp. 243–250.

375. D. Coppersmith, "Attack on the Cryptographic Scheme NIKS-TAS," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994, pp. 294–307.

376. D. Coppersmith, personal communication, 1994.

377. D. Coppersmith and E. Grossman, "Generators for Certain Alternating Groups with Applications to Cryptography," *SIAM Journal on Applied Mathematics*, v. 29, n. 4, Dec 1975, pp. 624–627.

378. D. Coppersmith, H. Krawczyk, and Y. Mansour, "The Shrinking Generator," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 22–39.

379. D. Coppersmith, A. Odlykzo, and R. Schroeppel, "Discrete Logarithms in GF(*p*)," *Algorithmica*, v. 1, n. 1, 1986, pp. 1–16.

380. D. Coppersmith and P. Rogaway, "Software Efficient Pseudo Random Function and the Use Thereof for Encryption," U.S. Patent pending, 1995.

381. D. Coppersmith, J. Stern, and S. Vaudenay, "Attacks on the Birational Signature Schemes," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 435–443.

382. V. Cordonnier and J.-J. Quisquater, eds., *CARDIS '94—Proceedings of the First Smart Card Research and Advanced Application Conference*, Lille, France, 24–26 Oct 1994.

383. C. Couvreur and J.-J. Quisquater, "An Introduction to Fast Generation of Large Prime Numbers," *Philips Journal Research*, v. 37, n. 5–6, 1982, pp. 231–264.

384. C. Couvreur and J.-J. Quisquater, "An Introduction to Fast Generation of Large Prime Numbers," *Philips Journal Research*, v. 38, 1983, p. 77.
385. C. Coveyou and R.D. MacPherson, "Fourier Analysis of Uniform Random Number Generators," *Journal of the ACM*, v. 14, n. 1, 1967, pp. 100–119.
386. T.M. Cover and R.C. King, "A Convergent Gambling Estimate of the Entropy of English," *IEEE Transactions on Information Theory*, v. IT-24, n. 4, Jul 1978, pp. 413–421.
387. R.J.F. Cramer and T.P. Pedersen, "Improved Privacy in Wallets with Observers," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 329–343.
388. R.E. Crandell, "Method and Apparatus for Public Key Exchange in a Cryptographic System," U.S. Patent #5,159,632, 27 Oct 1992.
389. C. Crépeau, "A Secure Poker Protocol That Minimizes the Effect of Player Coalitions," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 73–86.
390. C. Crépeau, "A Zero-Knowledge Poker Protocol that Achieves Confidentiality of the Players' Strategy, or How to Achieve an Electronic Poker Face," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 239–247.
391. C. Crépeau, "Equivalence Between Two Flavours of Oblivious Transfer," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 350–354.
392. C. Crépeau, "Correct and Private Reductions among Oblivious Transfers," Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1990.
393. C. Crépeau, "Quantum Oblivious Transfer," *Journal of Modern Optics*, v. 41, n. 12, Dec 1994, pp. 2445–2454.
394. C. Crépeau and J. Kilian, "Achieving Oblivious Transfer Using Weakened Security Assumptions," *Proceedings of the 29th Annual Symposium on the Foundations of Computer Science*, 1988, pp. 42–52.
395. C. Crépeau and J. Kilian, "Weakening Security Assumptions and Oblivious Transfer," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 2–7.
396. C. Crépeau and L. Salvail, "Quantum Oblivious Mutual Identification," *Advances in Cryptology—EUROCRYPT '95 Proceedings*, Springer-Verlag, 1995, pp. 133–146.
397. A. Curiger, H. Bonnenberg, R. Zimmermann, N. Felber, H. Kaeslin and W. Fichtner, "VINCI: VLSI Implementation of the New Block Cipher IDEA," *Proceedings of IEEE CICC '93*, San Diego, CA, May 1993, pp. 15.5.1–15.5.4.
398. A. Curiger and B. Stuber, "Specification for the IDEA Chip," Technical Report No. 92/03, Institut für Integrierte Systeme, ETH Zurich, Feb 1992.
399. T. Cusick, "Boolean Functions Satisfying a Higher Order Strict Avalanche Criterion," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 102–117.
400. T.W. Cusick and M.C. Wood, "The REDOC-II Cryptosystem," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 545–563.
401. Cylink Corporation, Cylink Corporation vs. RSA Data Security, Inc., Civil Action No. C94-02332-CW, United States District Court for the Northern District of California, 30 Jun 1994.
402. J. Daeman, "Cipher and Hash Function Design," Ph.D. Thesis, Katholieke Universiteit Leuven, Mar 95.
403. J. Daeman, A. Bosselaers, R. Govaerts, and J. Vandewalle, "Collisions for Schnorr's Hash Function FFT-Hash Presented at Crypto '91," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 477–480.
404. J. Daeman, R. Govaerts, and J. Vandewalle, "A Framework for the Design of One-Way Hash Functions Including Cryptanalysis of Damgård's One-Way Function Based on Cellular Automata," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 82–96.
405. J. Daeman, R. Govaerts, and J. Vandewalle, "A Hardware Design Model for Cryptographic Algorithms," *ESORICS 92, Proceedings of the Second European Symposium on Research in Computer Security*, Springer-Verlag, 1992, pp. 419–434.
406. J. Daemen, R. Govaerts, and J. Vandewalle, "Block Ciphers Based on Modular Arith-

metic," *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography,* Rome, Italy, 15–16 Feb 1993, pp. 80–89.

407. J. Daemen, R. Govaerts, and J. Vandewalle, "Fast Hashing Both in Hardware and Software," presented at the rump session of CRYPTO '93, Aug 1993.

408. J. Daeman, R. Govaerts, and J. Vandewalle, "Resynchronization Weaknesses in Synchronous Stream Ciphers," *Advances in Cryptology—EUROCRYPT '93 Proceedings,* Springer-Verlag, 1994, pp. 159–167.

409. J. Daeman, R. Govaerts, and J. Vandewalle, "Weak Keys for IDEA," *Advances in Cryptology—CRYPTO '93 Proceedings,* Springer-Verlag, 1994, pp. 224–230.

410. J. Daemen, R. Govaerts, and J. Vandewalle, "A New Approach to Block Cipher Design," *Fast Software Encryption, Cambridge Security Workshop Proceedings,* Springer-Verlag, 1994, pp. 18–32.

411. Z.-D. Dai, "Proof of Rueppel's Linear Complexity Conjecture," *IEEE Transactions on Information Theory,* v. IT-32, n. 3, May 1986, pp. 440–443.

412. I.B. Damgård, "Collision Free Hash Functions and Public Key Signature Schemes," *Advances in Cryptology—EUROCRYPT '87 Proceedings,* Springer-Verlag, 1988, pp. 203–216.

413. I.B. Damgård, "Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals," *Advances in Cryptology—CRYPTO '88 Proceedings,* Springer-Verlag, 1990, pp. 328–335.

414. I.B. Damgård, "A Design Principle for Hash Functions," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 416–427.

415. I.B. Damgård, "Practical and Provably Secure Release of a Secret and Exchange of Signatures," *Advances in Cryptology—EUROCRYPT '93 Proceedings,* Springer-Verlag, 1994, pp. 200–217.

416. I.B. Damgård and L.R. Knudsen, "The Breaking of the AR Hash Function," *Advances in Cryptology—EUROCRYPT '93 Proceedings,* Springer-Verlag, 1994, pp. 286–292.

417. I.B. Damgård and P. Landrock, "Improved Bounds for the Rabin Primality Test," *Cryptography and Coding III,* M.J. Ganley, ed., Oxford: Clarendon Press, 1993, pp. 117–128.

418. I.B. Damgård, P. Landrock and C. Pomerance, "Average Case Error Estimates for the Strong Probable Prime Test," *Mathematics of Computation,* v. 61, n. 203, Jul 1993, pp. 177–194.

419. H.E. Daniels, Jr., letter to Datapro Research Corporation regarding CCEP, 23 Dec 1985.

420. H. Davenport, *The Higher Arithmetic,* Dover Books, 1983.

421. G.I. Davida, "Inverse of Elements of a Galois Field," *Electronics Letters,* v. 8, n. 21, 19 Oct 1972. pp. 518–520.

422. G.I. Davida, "Hellman's Scheme Breaks DES in Its Basic Form," *IEEE Spectrum,* v. 16, n. 7, Jul 1979, p. 39.

423. G.I. Davida, "Chosen Signature Cryptanalysis of the RSA (MIT) Public Key Cryptosystem," *Technical Report TR-CS-82-2,* Department of EECS, University of Wisconsin, 1982.

424. G.I. Davida and G.G. Walter, "A Public Key Analog Cryptosystem," *Advances in Cryptology—EUROCRYPT '87 Proceedings,* Springer-Verlag, 1988, pp. 143–147.

425. G.I. Davida, D. Wells, and J. Kam, "A Database Encryption System with Subkeys," *ACM Transactions on Database Systems,* v. 6, n. 2, Jun 1981, pp. 312–328.

426. D.W. Davies, "Applying the RSA Digital Signature to Electronic Mail," *Computer,* v. 16, n. 2, Feb 1983, pp. 55–62.

427. D.W. Davies, "Some Regular Properties of the DES," *Advances in Cryptology: Proceedings of Crypto 82,* Plenum Press, 1983, pp. 89–96.

428. D.W. Davies, "A Message Authentication Algorithm Suitable for a Mainframe Computer," *Advances in Cryptology: Proceedings of Crypto 82,* Springer-Verlag, 1985, pp. 393–400.

429. D.W. Davies and S. Murphy, "Pairs and Triplets of DES S-boxes," *Cryptologia,* v. 8, n. 1, 1995, pp. 1–25.

430. D.W. Davies and G.I.P. Parkin, "The Average Size of the Key Stream in Output Feedback Encipherment," *Cryptography, Proceedings of the Workshop on Cryptography, Burg Feuerstein, Germany, March 29–April 2, 1982,* Springer-Verlag, 1983, pp. 263–279.

431. D.W. Davies and G.I.P. Parkin, "The Average Size of the Key Stream in Output Feed-

back Mode," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 97–98.

432. D.W. Davies and W.L. Price, "The Application of Digital Signatures Based on Public-Key Cryptosystems," *Proceedings of the Fifth International Computer Communications Conference*, Oct 1980, pp. 525–530.
433. D.W. Davies and W.L. Price, "The Application of Digital Signatures Based on Public-Key Cryptosystems," National Physical Laboratory Report DNACS 39/80, Dec 1980.
434. D.W. Davies and W.L. Price, "Digital Signature—An Update," *Proceedings of International Conference on Computer Communications, Sydney, Oct 1984*, North Holland: Elsevier, 1985, pp. 843–847.
435. D.W. Davies and W.L. Price, *Security for Computer Networks*, second edition, John Wiley & Sons, 1989.
436. M. Davio, Y. Desmedt, M. Fosseprez, R. Govaerts, J. Hulsbrosch, P. Neutjens, P. Piret, J.-J. Quisquater, J. Vandewalle, and S. Wouters, "Analytical Characteristics of the Data Encryption Standard," *Advances in Cryptology: Proceedings of Crypto 83*, Plenum Press, 1984, pp. 171–202.
437. M. Davio, Y. Desmedt, J. Goubert, F. Hoornaert, and J.-J. Quisquater, "Efficient Hardware and Software Implementation of the DES," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 144–146.
438. M. Davio, Y. Desmedt, and J.-J. Quisquater, "Propagation Characteristics of the DES," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, 62–73.
439. D. Davis, R. Ihaka, and P. Fenstermacher, "Cryptographic Randomness from Air Turbulence in Disk Drives," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994, pp. 114–120.
440. J.A. Davis, D.B. Holdbridge, and G.J. Simmons, "Status Report on Factoring (at the Sandia National Laboratories)," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 183–215.
441. R.M. Davis, "The Data Encryption Standard in Perspective," *Computer Security and the Data Encryption Standard*, National Bureau of Standards Special Publication 500-27, Feb 1978.
442. E. Dawson and A. Clark, "Cryptanalysis of Universal Logic Sequences," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, to appear.
443. M.H. Dawson and S.E. Tavares, "An Expanded Set of Design Criteria for Substitution Boxes and Their Use in Strengthening DES-Like Cryptosystems," *IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, Victoria, BC, Canada, 9–10 May 1991, pp. 191–195.
444. M.H. Dawson and S.E. Tavares, "An Expanded Set of S-Box Design Criteria Based on Information Theory and Its Relation to Differential-like Attacks," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 352–367.
445. C.A. Deavours, "Unicity Points in Cryptanalysis," *Cryptologia*, v. 1, n. 1, 1977, pp. 46–68.
446. C.A. Deavours, "The Black Chamber: A Column; How the British Broke Enigma," *Cryptologia*, v. 4, n. 3, Jul 1980, pp. 129–132.
447. C.A. Deavours, "The Black Chamber: A Column; La Méthode des Bâtons," *Cryptologia*, v. 4, n. 4, Oct 1980, pp. 240–247.
448. C.A. Deavours and L. Kruh, *Machine Cryptography and Modern Cryptanalysis*, Norwood MA: Artech House, 1985.
449. J.M. DeLaurentis, "A Further Weakness in the Common Modulus Protocol for the RSA Cryptosystem," *Cryptologia*, v. 8, n. 3, Jul 1984, pp. 253–259.
450. P. Delsarte, Y. Desmedt, A. Odlyzko, and P. Piret, "Fast Cryptanalysis of the Matsumoto-Imai Public-Key Scheme," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 142–149.
451. P. Delsarte and P. Piret, "Comment on 'Extension of RSA Cryptostructure: A Galois Approach'," *Electronics Letters*, v. 18, n. 13, 24 Jun 1982, pp. 582–583.
452. R. DeMillo, N. Lynch, and M. Merritt, "Cryptographic Protocols," *Proceedings of the 14th Annual Symposium on the Theory of Computing*, 1982, pp. 383–400.
453. R. DeMillo and M. Merritt, "Protocols for Data Security," *Computer*, v. 16, n. 2, Feb 1983, pp. 39–50.
454. N. Demytko, "A New Elliptic Curve Based Analogue of RSA," *Advances in Cryptol-*

ogy—EUROCRYPT '93 Proceedings, Springer-Verlag, 1994, pp. 40–49.

455. D.E. Denning, "Secure Personal Computing in an Insecure Network," *Communications of the ACM*, v. 22, n. 8, Aug 1979, pp. 476–482.
456. D.E. Denning, *Cryptography and Data Security*, Addison-Wesley, 1982.
457. D.E. Denning, "Protecting Public Keys and Signature Keys," *Computer*, v. 16, n. 2, Feb 1983, pp. 27–35.
458. D.E. Denning, "Digital Signatures with RSA and Other Public-Key Cryptosystems," *Communications of the ACM*, v. 27, n. 4, Apr 1984, pp. 388–392.
459. D.E. Denning, "The Data Encryption Standard: Fifteen Years of Public Scrutiny," *Proceedings of the Sixth Annual Computer Security Applications Conference*, IEEE Computer Society Press, 1990.
460. D.E. Denning, "The Clipper Chip: A Technical Summary," unpublished manuscript, 21 Apr 1993.
461. D.E. Denning and G.M. Sacco, "Timestamps in Key Distribution Protocols," *Communications of the ACM*, v. 24, n. 8, Aug 1981, pp. 533–536.
462. D.E. Denning and M. Smid, "Key Escrowing Today," *IEEE Communications Magazine*, v. 32, n. 9, Sep 1994, pp. 58–68.
463. T. Denny, B. Dodson, A.K. Lenstra, and M.S. Manasse, "On the Factorization of RSA-120," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 166–174.
464. W.F. Denny, "Encryptions Using Linear and Non-Linear Codes: Implementations and Security Considerations," Ph.D. dissertation, The Center for Advanced Computer Studies, University of Southern Louisiana, Spring 1988.
465. Department of Defense, "Department of Defense Trusted Computer System Evaluation Criteria," DOD 5200.28-STD, Dec 1985.
466. Department of State, "International Traffic in Arms Regulations (ITAR)," 22 CFR 120–130, Office of Munitions Control, Nov 1989.
467. Department of State, "Defense Trade Regulations," 22 CFR 120–130, Office of Defense Trade Controls, May 1992.
468. Department of the Treasury, "Electronic Funds and Securities Transfer Policy," Department of the Treasury Directives Manual, Chapter TD 81, Section 80, Department of the Treasury, 16 Aug 1984.
469. Department of the Treasury, "Criteria and Procedures for Testing, Evaluating, and Certifying Message Authentication Decisions for Federal E.F.T. Use," Department of the Treasury, 1 May 1985.
470. Department of the Treasury, "Electronic Funds and Securities Transfer Policy—Message Authentication and Enhanced Security," Order No. 106-09, Department of the Treasury, 2 Oct 1986.
471. H. Dobbertin, "A Survey on the Construction of Bent Functions," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
472. B. Dodson and A.K. Lenstra, "NFS with Four Large Primes: An Explosive Experiment," draft manuscript.
473. D. Dolev and A. Yao, "On the Security of Public-Key Protocols," *Communications of the ACM*, v. 29, n. 8, Aug 1983, pp. 198–208.
474. J. Domingo-Ferrer, "Probabilistic Authentication Analysis," *CARDIS 94—Proceedings of the First Smart Card Research and Applications Conference*, Lille, France, 24–26 Oct 1994, pp. 49–60.
475. P. de Rooij, "On the Security of the Schnorr Scheme Using Preprocessing," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 71–80.
476. A. De Santis, G. Di Crescenzo, and G. Persiano, "Secret Sharing and Perfect Zero Knowledge," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 73–84.
477. A. De Santis, S. Micali, and G. Persiano, "Non-Interactive Zero-Knowledge Proof Systems," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 52–72.
478. A. De Santis, S. Micali, and G. Persiano, "Non-Interactive Zero-Knowledge with Preprocessing," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 269–282.
479. Y. Desmedt, "What Happened with Knapsack Cryptographic Schemes" *Performance Limits in Communication, Theory and Practice*, NATO ASI Series E: Applied Sciences, v. 142, Kluwer Academic Publishers, 1988, pp. 113–134.
480. Y. Desmedt, "Subliminal-Free Authentication and Signature," *Advances in Cryptol-*

ogy—EUROCRYPT '88 Proceedings, Springer-Verlag, 1988, pp. 23–33.

481. Y. Desmedt, "Abuses in Cryptography and How to Fight Them," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 375–389.
482. Y. Desmedt and M. Burmester, "An Efficient Zero-Knowledge Scheme for the Discrete Logarithm Based on Smooth Numbers," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 360–367.
483. Y. Desmedt and Y. Frankel, "Threshold Cryptosystems," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 307–315.
484. Y. Desmedt and Y. Frankel, "Shared Generation of Authentication and Signatures," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 457–469.
485. Y. Desmedt, C. Goutier, and S. Bengio, "Special Uses and Abuses of the Fiat-Shamir Passport Protocol," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 21–39.
486. Y. Desmedt and A.M. Odlykzo, "A Chosen Text Attack on the RSA Cryptosystem and Some Discrete Logarithm Problems," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 516–522.
487. Y. Desmedt, J.-J. Quisquater, and M. Davio, "Dependence of Output on Input in DES: Small Avalanche Characteristics," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 359–376.
488. Y. Desmedt, J. Vandewalle, and R. Govaerts, "Critical Analysis of the Security of Knapsack Public Key Algorithms," *IEEE Transactions on Information Theory*, v. IT-30, n. 4, Jul 1984, pp. 601–611.
489. Y. Desmedt and M. Yung, "Weaknesses of Undeniable Signature Schemes," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 205–220.
490. W. Diffie, lecture at IEEE Information Theory Workshop, Ithaca, N.Y., 1977.
491. W. Diffie, "Cryptographic Technology: Fifteen Year Forecast," BNR Inc., Jan 1981.
492. W. Diffie, "The First Ten Years of Public-Key Cryptography," *Proceedings of the IEEE*, v. 76, n. 5, May 1988, pp. 560–577.
493. W. Diffie, "Authenticated Key Exchange and Secure Interactive Communication," *Proceedings of SECURICOM '90*, 1990.
494. W. Diffie, "The First Ten Years of Public-Key Cryptography," in *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, pp. 135–175.
495. W. Diffie and M.E. Hellman, "Multiuser Cryptographic Techniques," *Proceedings of AFIPS National Computer Conference*, 1976, pp. 109–112.
496. W. Diffie and M.E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, v. IT-22, n. 6, Nov 1976, pp. 644–654.
497. W. Diffie and M.E. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer*, v. 10, n. 6, Jun 1977, pp. 74–84.
498. W. Diffie and M.E. Hellman, "Privacy and Authentication: An Introduction to Cryptography," *Proceedings of the IEEE*, v. 67, n. 3, Mar 1979, pp. 397–427.
499. W. Diffie, L. Strawczynski, B. O'Higgins, and D. Steer, "An ISDN Secure Telephone Unit," *Proceedings of the National Telecommunications Forum*, v. 41, n. 1, 1987, pp. 473–477.
500. W. Diffie, P.C. van Oorschot, and M.J. Wiener, "Authentication and Authenticated Key Exchanges," *Designs, Codes and Cryptography*, v. 2, 1992, 107–125.
501. C. Ding, "The Differential Cryptanalysis and Design of Natural Stream Ciphers," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 101–115.
502. C. Ding, G. Xiao, and W. Shan, *The Stability Theory of Stream Ciphers*, Springer-Verlag, 1991.
503. A. Di Porto and W. Wolfowicz, "VINO: A Block Cipher Including Variable Permutations," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 205–210.
504. B. Dixon and A.K. Lenstra, "Factoring Integers Using SIMD Sieves," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 28–39.
505. J.D. Dixon, "Factorization and Primality Tests," *American Mathematical Monthly*, v. 91, n. 6, 1984, pp. 333–352.
506. D. Dolev and A. Yao, "On the Security of Public Key Protocols," *Proceedings of the*

22nd Annual Symposium on the Foundations of Computer Science, 1981, pp. 350–357.

507. L.X. Duan and C.C. Nian, "Modified Lu-Lee Cryptosystems," *Electronics Letters,* v. 25, n. 13, 22 Jun 1989, p. 826.
508. R. Durstenfeld, "Algorithm 235: Random Permutation," *Communications of the ACM,* v. 7, n. 7, Jul 1964, p. 420.
509. S. Dussé and B. Kaliski, Jr., "A Cryptographic Library for the Motorola DSP56000," *Advances in Cryptology—EUROCRYPT '90 Proceedings,* Springer-Verlag, 1991, pp. 230–244.
510. C. Dwork and L. Stockmeyer, "Zero-Knowledge with Finite State Verifiers," *Advances in Cryptology—CRYPTO '88 Proceedings,* Springer-Verlag, 1990, pp. 71–75.
511. D.E. Eastlake, S.D. Crocker, and J.I. Schiller, "Randomness Requirements for Security," RFC 1750, Dec 1994.
512. H. Eberle, "A High-Speed DES Implementation for Network Applications," *Advances in Cryptology—CRYPTO '92 Proceedings,* Springer-Verlag, pp. 521–539.
513. J. Edwards, "Implementing Electronic Poker: A Practical Exercise in Zero-Knowledge Interactive Proofs," Master's thesis, Department of Computer Science, University of Kentucky, May 1994.
514. W.F. Ehrsam, C.H.W. Meyer, R.L. Powers, J.L. Smith, and W.L. Tuchman, "Product Block Cipher for Data Security," U.S. Patent #3,962,539, 8 Jun 1976.
515. W.F. Ehrsam, C.H.W. Meyer, and W.L. Tuchman, "A Cryptographic Key Management Scheme for Implementing the Data Encryption Standard," *IBM Systems Journal,* v. 17, n. 2, 1978, pp. 106–125.
516. R. Eier and H. Lagger, "Trapdoors in Knapsack Cryptosystems," *Lecture Notes in Computer Science 149; Cryptography—Proceedings, Burg Feuerstein 1982,* Springer-Verlag, 1983, pp. 316–322.
517. A.K. Ekert, "Quantum Cryptography Based on Bell's Theorem," *Physical Review Letters,* v. 67, n. 6, Aug 1991, pp. 661–663.
518. T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *Advances in Cryptology: Proceedings of CRYPTO 84,* Springer-Verlag, 1985, pp. 10–18.
519. T. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory,* v. IT-31, n. 4, 1985, pp. 469–472.
520. T. ElGamal, "On Computing Logarithms Over Finite Fields," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 396–402.
521. T. ElGamal and B. Kaliski, letter to the editor regarding LUC, *Dr. Dobb's Journal,* v. 18, n. 5, May 1993, p. 10.
522. T. Eng and T. Okamoto, "Single-Term Divisible Electronic Coins," *Advances in Cryptology—EUROCRYPT '94 Proceedings,* Springer-Verlag, 1995, to appear.
523. M.H. Er, D.J. Wong, A.A. Sethu, and K.S. Ngeow, "Design and Implementation of RSA Cryptosystem Using Multiple DSP Chips," *1991 IEEE International Symposium on Circuits and Systems,* v. 1, Singapore, 11–14 Jun 1991, pp. 49–52.
524. D. Estes, L.M. Adleman, K. Konpella, K.S. McCurley, and G.L. Miller, "Breaking the Ong-Schnorr-Shamir Signature Schemes for Quadratic Number Fields," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 3–13.
525. ETEBAC, "Échanges Télématiques Entre Les Banques et Leurs Clients," Standard ETEBAC 5, *Comité Français d'Organisation et de Normalisation Bancaires,* Apr 1989. (In French.)
526. A. Evans, W. Kantrowitz, and E. Weiss, "A User Identification Scheme Not Requiring Secrecy in the Computer," *Communications of the ACM,* v. 17, n. 8, Aug 1974, pp. 437–472.
527. S. Even and O. Goldreich, "DES-Like Functions Can Generate the Alternating Group," *IEEE Transactions on Information Theory,* v. IT-29, n. 6, Nov 1983, pp. 863–865.
528. S. Even and O. Goldreich, "On the Power of Cascade Ciphers," *ACM Transactions on Computer Systems,* v. 3, n. 2, May 1985, pp. 108–116.
529. S. Even, O. Goldreich, and A. Lempel, "A Randomizing Protocol for Signing Contracts," *Communications of the ACM,* v. 28, n. 6, Jun 1985, pp. 637–647.
530. S. Even and Y. Yacobi, "Cryptography and **NP**-Completeness," *Proceedings of the 7th International Colloquium on Automata,*

Languages, and Programming, Springer-Verlag, 1980, pp. 195–207.

531. H.-H. Evertse, "Linear Structures in Block Ciphers," *Advances in Cryptology—EUROCRYPT '87 Proceedings,* Springer-Verlag, 1988, pp. 249–266.
532. P. Fahn and M.J.B. Robshaw, "Results from the RSA Factoring Challenge," Technical Report TR-501, Version 1.3, RSA Laboratories, Jan 1995.
533. R.C. Fairfield, A. Matusevich, and J. Plany, "An LSI Digital Encryption Processor (DEP)," *Advances in Cryptology: Proceedings of CRYPTO 84,* Springer-Verlag, 1985, pp. 115–143.
534. R.C. Fairfield, A. Matusevich, and J. Plany, "An LSI Digital Encryption Processor (DEP)," *IEEE Communications,* v. 23, n. 7, Jul 1985, pp. 30–41.
535. R.C. Fairfield, R.L. Mortenson, and K.B. Koulthart, "An LSI Random Number Generator (RNG)," *Advances in Cryptology: Proceedings of CRYPTO 84,* Springer-Verlag, 1985, pp. 203–230.
536. "International Business Machines Corp. License Under Patents," *Federal Register,* v. 40, n. 52, 17 Mar 1975, p. 12067.
537. "Solicitation for Public Key Cryptographic Algorithms," *Federal Register,* v. 47, n. 126, 30 Jun 1982, p. 28445.
538. "Proposed Federal Information Processing Standard for Digital Signature Standard (DSS)," *Federal Register,* v. 56, n. 169, 30 Aug 1991, pp. 42980–42982.
539. "Proposed Federal Information Processing Standard for Secure Hash Standard," *Federal Register,* v. 57, n. 21, 31 Jan 1992, pp. 3747–3749.
540. "Proposed Reaffirmation of Federal Information Processing Standard (FIPS) 46-1, Data Encryption Standard (DES)," *Federal Register,* v. 57, n. 177, 11 Sep 1992, p. 41727.
541. "Notice of Proposal for Grant of Exclusive Patent License," *Federal Register,* v. 58, n. 108, 8 Jun 1993, pp. 23105–23106.
542. "Approval of Federal Information Processing Standards Publication 186, Digital Signature Standard (DSS)," *Federal Register,* v. 58, n. 96, 19 May 1994, pp. 26208–26211.
543. "Proposed Revision of Federal Information Processing Standard (FIPS) 180, Secure Hash Standard," *Federal Register,* v. 59, n. 131, 11 Jul 1994, pp. 35317–35318.
544. U. Feige, A. Fiat, and A. Shamir, "Zero Knowledge Proofs of Identity," *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing,* 1987, pp. 210–217.
545. U. Feige, A. Fiat, and A. Shamir, "Zero Knowledge Proofs of Identity," *Journal of Cryptology,* v. 1, n. 2, 1988, pp. 77–94.
546. U. Feige and A. Shamir, "Zero Knowledge Proofs of Knowledge in Two Rounds," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 526–544.
547. J. Feigenbaum, "Encrypting Problem Instances, or, . . . , Can You Take Advantage of Someone Without Having to Trust Him," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 477–488.
548. J. Feigenbaum, "Overview of Interactive Proof Systems and Zero-Knowledge," in *Contemporary Cryptology: The Science of Information Integrity,* G.J. Simmons, ed., IEEE Press, 1992, pp. 423–439.
549. J. Feigenbaum, M.Y. Liberman, E. Grosse, and J.A. Reeds, "Cryptographic Protection of Membership Lists," *Newsletter of the International Association of Cryptologic Research,* v. 9, 1992, pp. 16–20.
550. J. Feigenbaum, M.Y. Liverman, and R.N. Wright, "Cryptographic Protection of Databases and Software," *Distributed Computing and Cryptography,* J. Feigenbaum and M. Merritt, eds., American Mathematical Society, 1991, pp. 161–172.
551. H. Feistel, "Cryptographic Coding for Data-Bank Privacy," RC 2827, Yorktown Heights, NY: IBM Research, Mar 1970.
552. H. Feistel, "Cryptography and Computer Privacy," *Scientific American,* v. 228, n. 5, May 1973, pp. 15–23.
553. H. Feistel, "Block Cipher Cryptographic System," U.S. Patent #3,798,359, 19 Mar 1974.
554. H. Feistel, "Step Code Ciphering System," U.S. Patent #3,798,360, 19 Mar 1974.
555. H. Feistel, "Centralized Verification System," U.S. Patent #3,798,605, 19 Mar 1974.
556. H. Feistel, W.A. Notz, and J.L. Smith, "Cryptographic Techniques for Machine to Machine Data Communications," RC 3663, Yorktown Heights, N.Y.: IBM Research, Dec 1971.

557. H. Feistel, W.A. Notz, and J.L. Smith, "Some Cryptographic Techniques for Machine to Machine Data Communications," *Proceedings of the IEEE*, v. 63, n. 11, Nov 1975, pp. 1545–1554.

558. P. Feldman, "A Practical Scheme for Non-interactive Verifiable Secret Sharing," *Proceedings of the 28th Annual Symposium on the Foundations of Computer Science*, 1987, pp. 427–437.

559. R.A. Feldman, "Fast Spectral Test for Measuring Nonrandomness and the DES," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 243–254.

560. R.A. Feldman, "A New Spectral Test for Nonrandomness and the DES," *IEEE Transactions on Software Engineering*, v. 16, n. 3, Mar 1990, pp. 261–267.

561. D.C. Feldmeier and P.R. Karn, "UNIX Password Security—Ten Years Later," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 44–63.

562. H. Fell and W. Diffie, "Analysis of a Public Key Approach Based on Polynomial Substitution," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 427–437.

563. N.T. Ferguson, "Single Term Off-Line Coins," Report CS-R9318, Computer Science/Department of Algorithms and Architecture, CWI, Mar 1993.

564. N.T. Ferguson, "Single Term Off-Line Coins," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 318–328.

565. N.T. Ferguson, "Extensions of Single-term Coins," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 292–301.

566. A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 186–194.

567. A. Fiat and A. Shamir, "Unforgeable Proofs of Identity," *Proceedings of Securicom 87*, Paris, 1987, pp. 147–153.

568. P. Finch, "A Study of the Blowfish Encryption Algorithm," Ph.D. dissertation, Department of Computer Science, City University of New York Graduate School and University Center, Feb 1995.

569. R. Flynn and A.S. Campasano, "Data Dependent Keys for Selective Encryption Terminal," *Proceedings of NCC, vol. 47*, AFIPS Press, 1978, pp. 1127–1129.

570. R.H. Follett, letter to NIST regarding DSS, 25 Nov 1991.

571. R. Forré, "The Strict Avalanche Criterion: Spectral Properties and an Extended Definition," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 450–468.

572. R. Forré, "A Fast Correlation Attack on Nonlinearity Feedforward Filtered Shift Register Sequences," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 568–595.

573. S. Fortune and M. Merritt, "Poker Protocols," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 454–464.

574. R.B. Fougner, "Public Key Standards and Licenses," RFC 1170, Jan 1991.

575. Y. Frankel and M. Yung, "Escrowed Encryption Systems Visited: Threats, Attacks, Analysis and Designs," *Advances in Cryptology—CRYPTO '95 Proceedings*, Springer-Verlag, 1995, to appear.

576. W.F. Friedman, *Methods for the Solution of Running-Key Ciphers*, Riverbank Publication No. 16, Riverbank Labs, 1918.

577. W.F. Friedman, *The Index of Coincidence and Its Applications in Cryptography*, Riverbank Publication No. 22, Riverbank Labs, 1920. Reprinted by Aegean Park Press, 1987.

578. W.F. Friedman, *Elements of Cryptanalysis*, Laguna Hills, CA: Aegean Park Press, 1976.

579. W.F. Friedman, "Cryptology," *Encyclopedia Britannica*, v. 6, pp. 844–851, 1967.

580. A.M. Frieze, J. Hastad, R. Kannan, J.C. Lagarias, and A. Shamir, "Reconstructing Truncated Integer Variables Satisfying Linear Congruences," *SIAM Journal on Computing*, v. 17, n. 2, Apr 1988, pp. 262–280.

581. A.M. Frieze, R. Kannan, and J.C. Lagarias, "Linear Congruential Generators Do not Produce Random Sequences," *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science*, 1984, pp. 480–484.

582. E. Fujiaski and T. Okamoto, "On Comparison of Practical Digitial Signature Schemes," *Proceedings of the 1992 Sym-*

posium on Cryptography and Information Security (SCIS 92), Tateshina, Japan, 2–4 Apr 1994, pp. 1A.1–12.

583. A. Fujioka, T. Okamoto, and S. Miyaguchi, "ESIGN: An Efficient Digital Signature Implementation for Smart Cards," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 446–457.
584. A. Fujioka, T. Okamoto, and K. Ohta, "Interactive Bi-Proof Systems and Undeniable Signature Schemes," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 243–256.
585. A. Fujioka, T. Okamoto, and K. Ohta, "A Practical Secret Voting Scheme for Large Scale Elections," *Advances in Cryptology—AUSCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 244–251.
586. K. Gaardner and E. Snekkenes, "Applying a Formal Analysis Technique to the CCITT X.509 Strong Two-Way Authentication Protocol," *Journal of Cryptology*, v. 3, n. 2, 1991, pp. 81–98.
587. H.F. Gaines, *Cryptanalysis*, American Photographic Press, 1937. (Reprinted by Dover Publications, 1956.)
588. J. Gait, "A New Nonlinear Pseudorandom Number Generator," *IEEE Transactions on Software Engineering*, v. SE-3, n. 5, Sep 1977, pp. 359–363.
589. J. Gait, "Short Cycling in the Kravitz-Reed Public Key Encryption System," *Electronics Letters*, v. 18, n. 16, 5 Aug 1982, pp. 706–707.
590. Z. Galil, S. Haber, and M. Yung, "A Private Interactive Test of a Boolean Predicate and Minimum-Knowledge Public-Key Cryptosystems," *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, 1985, pp. 360–371.
591. Z. Galil, S. Haber, and M. Yung, "Cryptographic Computation: Secure Fault-Tolerant Protocols and the Public-Key Model," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 135–155.
592. Z. Galil, S. Haber, and M. Yung, "Minimum-Knowledge Interactive Proofs for Decision Problems," *SIAM Journal on Computing*, v. 18, n. 4, 1989, pp. 711–739.
593. R.G. Gallager, *Information Theory and Reliable Communications*, New York: John Wiley & Sons, 1968.
594. P. Gallay and E. Depret, "A Cryptography Microprocessor," *1988 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 1988, pp. 148–149.
595. R.A. Games, "There are no de Bruijn Sequences of Span n with Complexity $2^{n-1} + n + 1$," *Journal of Combinatorical Theory*, Series A, v. 34, n. 2, Mar 1983, pp. 248–251.
596. R.A. Games and A.H. Chan, "A Fast Algorithm for Determining the Complexity of a Binary Sequence with 2^n," *IEEE Transactions on Information Theory*, v. IT-29, n. 1, Jan 1983, pp. 144–146.
597. R.A. Games, A.H. Chan, and E.L. Key, "On the Complexity of de Bruijn Sequences," *Journal of Combinatorical Theory*, Series A, v. 33, n. 1, Nov 1982, pp. 233–246.
598. S.H. Gao and G.L. Mullen, "Dickson Polynomials and Irreducible Polynomials over Finite Fields," *Journal of Number Theory*, v. 49, n. 1, Oct 1994, pp. 18–132.
599. M. Gardner, "A New Kind of Cipher That Would Take Millions of Years to Break," *Scientific American*, v. 237, n. 8, Aug 1977, pp. 120–124.
600. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of **NP**-Completeness*, W.H. Freeman and Co., 1979.
601. S.L. Garfinkel, *PGP: Pretty Good Privacy*, Sebastopol, CA: O'Reilly and Associates, 1995.
602. C.W. Gardiner, "Distributed Public Key Certificate Management," *Proceedings of the Privacy and Security Research Group 1993 Workshop on Network and Distributed System Security*, The Internet Society, 1993, pp. 69–73.
603. G. Garon and R. Outerbridge, "DES Watch: An Examination of the Sufficiency of the Data Encryption Standard for Financial Institution Information Security in the 1990's," *Cryptologia*, v. 15, n. 3, Jul 1991, pp. 177–193.
604. M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson, "The Digital Distributed Systems Security Architecture," *Proceedings of the 12th National Computer Security Conference*, NIST, 1989, pp. 305–319.
605. J. von zur Gathen, D. Kozen, and S. Landau, "Functional Decomposition of Polynomials," *Proceedings of the 28th IEEE Symposium on the Foundations of Com-*

puter Science, IEEE Press, 1987, pp. 127–131.

606. P.R. Geffe, "How to Protect Data With Ciphers That are Really Hard to Break," *Electronics*, v. 46, n. 1, Jan 1973, pp. 99–101.
607. D.K. Gifford, D. Heitmann, D.A. Segal, R.G. Cote, K. Tanacea, and D.E. Burmaster, "Boston Community Information System 1986 Experimental Test Results," MIT/LCS/TR-397, MIT Laboratory for Computer Science, Aug 1987.
608. D.K. Gifford, J.M. Lucassen, and S.T. Berlin, "The Application of Digital Broadcast Communication to Large Scale Information Systems," *IEEE Journal on Selected Areas in Communications*, v. 3, n. 3, May 1985, pp. 457–467.
609. D.K. Gifford and D.A. Segal, "Boston Community Information System 1987–1988 Experimental Test Results," MIT/LCS/TR-422, MIT Laboratory for Computer Science, May 1989.
610. H. Gilbert and G. Chase, "A Statistical Attack on the Feal-8 Cryptosystem," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 22–33.
611. H. Gilbert and P. Chauvaud, "A Chosen Plaintext Attack of the 16-Round Khufu Cryptosystem," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994, pp. 259–268.
612. M. Girault, "Hash-Functions Using Modulo-*N* Operations," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 217–226.
613. J. Gleick, "A New Approach to Protecting Secrets is Discovered," *The New York Times*, 18 Feb 1987, pp. C1 and C3.
614. J.-M. Goethals and C. Couvreur, "A Cryptanalytic Attack on the Lu-Lee Public-Key Cryptosystem," *Philips Journal of Research*, v. 35, 1980, pp. 301–306.
615. O. Goldreich, "A Uniform-Complexity Treatment of Encryption and Zero-Knowledge, *Journal of Cryptology*, v. 6, n. 1, 1993, pp. 21–53.
616. O. Goldreich and H. Krawczyk, "On the Composition of Zero Knowledge Proof Systems," *Proceedings on the 17th International Colloquium on Automata, Languages, and Programming*, Springer-Verlag, 1990, pp. 268–282.
617. O. Goldreich and E. Kushilevitz, "A Perfect Zero-Knowledge Proof for a Problem Equivalent to Discrete Logarithm," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 58–70.
618. O. Goldreich and E. Kushilevitz, "A Perfect Zero-Knowledge Proof for a Problem Equivalent to Discrete Logarithm," *Journal of Cryptology*, v. 6, n. 2, 1993, pp. 97–116.
619. O. Goldreich, S. Micali, and A. Wigderson, "Proofs That Yield Nothing but Their Validity and a Methodology of Cryptographic Protocol Design," *Proceedings of the 27th IEEE Symposium on the Foundations of Computer Science*, 1986, pp. 174–187.
620. O. Goldreich, S. Micali, and A. Wigderson, "How to Prove All **NP** Statements in Zero Knowledge and a Methodology of Cryptographic Protocol Design," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 171–185.
621. O. Goldreich, S. Micali, and A. Wigderson, "How to Play Any Mental Game," *Proceedings of the 19th ACM Symposium on the Theory of Computing*, 1987, pp. 218–229.
622. O. Goldreich, S. Micali, and A. Wigderson, "Proofs That Yield Nothing but Their Validity and a Methodology of Cryptographic Protocol Design," *Journal of the ACM*, v. 38, n. 1, Jul 1991, pp. 691–729.
623. S. Goldwasser and J. Kilian, "Almost All Primes Can Be Quickly Certified," *Proceedings of the 18th ACM Symposium on the Theory of Computing*, 1986, pp. 316–329.
624. S. Goldwasser and S. Micali, "Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information," *Proceedings of the 14th ACM Symposium on the Theory of Computing*, 1982, pp. 270–299.
625. S. Goldwasser and S. Micali, "Probabilistic Encryption," *Journal of Computer and System Sciences*, v. 28, n. 2, Apr 1984, pp. 270–299.
626. S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," *Proceedings of the 17th ACM Symposium on Theory of Computing*, 1985, pp. 291–304.

627. S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," *SIAM Journal on Computing*, v. 18, n. 1, Feb 1989, pp. 186–208.
628. S. Goldwasser, S. Micali, and R.L. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks," *SIAM Journal on Computing*, v. 17, n. 2, Apr 1988, pp. 281–308.
629. S. Goldwasser, S. Micali, and A.C. Yao, "On Signatures and Authentication," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 211–215.
630. J.D. Golič, "On the Linear Complexity of Functions of Periodic GF(q) Sequences," *IEEE Transactions on Information Theory*, v. IT-35, n. 1, Jan 1989, pp. 69–75.
631. J.D. Golič, "Linear Cryptanalysis of Stream Ciphers," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, pp. 262–282.
632. J.D. Golič, "Towards Fast Correlation Attacks on Irregularly Clocked Shift Registers," *Advances in Cryptology—EUROCRYPT '95 Proceedings*, Springer-Verlag, 1995, to appear.
633. J.D. Golič and M.J. Mihajlevič, "A Generalized Correlation Attack on a Class of Stream Ciphers Based on the Levenshtein Distance," *Journal of Cryptology*, v. 3, n. 3, 1991, pp. 201–212.
634. J.D. Golič and L. O'Connor, "Embedding and Probabilistic Correlation Attacks on Clock-Controlled Shift Registers," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.
635. R. Golliver, A.K. Lenstra, K.S. McCurley, "Lattice Sieving and Trial Division," *Proceedings of the Algorithmic Number Theory Symposium*, Cornell, 1994, to appear.
636. D. Gollmann, "Kaskadenschaltungen taktgesteuerter Schieberegister als Pseudozufallszahlengeneratoren," Ph.D. dissertation, Universität Linz, 1983. (In German.)
637. D. Gollmann, "Pseudo Random Properties of Cascade Connections of Clock Controlled Shift Registers," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 93–98.
638. D. Gollmann, "Correlation Analysis of Cascaded Sequences," *Cryptography and Coding*, H.J. Beker and F.C. Piper, eds., Oxford: Clarendon Press, 1989, pp. 289–297.
639. D. Gollmann, "Transformation Matrices of Clock-Controlled Shift Registers," *Cryptography and Coding III*, M.J. Ganley, ed., Oxford: Clarendon Press, 1993, pp. 197–210.
640. D. Gollmann and W.G. Chambers, "Lock-In Effect in Cascades of Clock-Controlled Shift-Registers," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 331–343.
641. D. Gollmann and W.G. Chambers, "Clock-Controlled Shift Registers: A Review," *IEEE Journal on Selected Areas in Communications*, v. 7, n. 4, May 1989, pp. 525–533.
642. D. Gollmann and W.G. Chambers, "A Cryptanalysis of Step$_{k,m}$-cascades," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 680–687.
643. S.W. Golomb, *Shift Register Sequences*, San Francisco: Holden-Day, 1967. (Reprinted by Aegean Park Press, 1982.)
644. L. Gong, "A Security Risk of Depending on Synchronized Clocks," *Operating Systems Review*, v. 26, n. 1, Jan 1992, pp. 49–53.
645. L. Gong, R. Needham, and R. Yahalom, "Reasoning About Belief in Cryptographic Protocols," *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 1991, pp. 234–248.
646. R.M. Goodman and A.J. McAuley, "A New Trapdoor Knapsack Public Key Cryptosystem," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 150–158.
647. R.M. Goodman and A.J. McAuley, "A New Trapdoor Knapsack Public Key Cryptosystem," *IEE Proceedings*, v. 132, pt. E, n. 6, Nov 1985, pp. 289–292.
648. D.M. Gordon, "Discrete Logarithms Using the Number Field Sieve," Preprint, 28 Mar 1991.
649. D.M. Gordon and K.S. McCurley, "Computation of Discrete Logarithms in Fields of Characteristic Two," presented at the rump session of CRYPTO '91, Aug 1991.
650. D.M. Gordon and K.S. McCurley, "Massively Parallel Computation of Discrete Logarithms," *Advances in Cryptology—*

CRYPTO '92 Proceedings, Springer-Verlag, 1993, pp. 312–323.

651. J.A. Gordon, "Strong Primes are Easy to Find," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 216–223.
652. J.A. Gordon, "Very Simple Method to Find the Minimal Polynomial of an Arbitrary Non-Zero Element of a Finite Field," *Electronics Letters*, v. 12, n. 25, 9 Dec 1976, pp. 663–664.
653. J.A. Gordon and R. Retkin, "Are Big S-Boxes Best?" *Cryptography, Proceedings of the Workshop on Cryptography, Burg Feuerstein, Germany, March 29–April 2, 1982*, Springer-Verlag, 1983, pp. 257–262.
654. M. Goresky and A. Klapper, "Feedback Registers Based on Ramified Extension of the 2-adic Numbers," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.
655. GOST, Gosudarstvennyi Standard 28147-89, "Cryptographic Protection for Data Processing Systems," Government Committee of the USSR for Standards, 1989. (In Russian.)
656. GOST R 34.10-94, Gosudarstvennyi Standard of Russian Federation, "Information technology. Cryptographic Data Security. Produce and check procedures of Electronic Digital Signature based on Asymmetric Cryptographic Algorithm." Government Committee of the Russia for Standards, 1994. (In Russian.)
657. GOST R 34.11-94, Gosudarstvennyi Standard of Russian Federation, "Information technology. Cryptographic Data Security. Hashing function." Government Committee of the Russia for Standards, 1994. (In Russian.)
658. R. Göttfert and H. Niederreiter, "On the Linear Complexity of Products of Shift-Register Sequences," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 151–158.
659. R. Göttfert and H. Niederreiter, "A General Lower Bound for the Linear Complexity of the Product of Shift-Register Sequences," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.
660. J. van de Graaf and R. Peralta, "A Simple and Secure Way to Show the Validity of Your Public Key," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 128–134.
661. J. Grollman and A.L. Selman, "Complexity Measures for Public-Key Cryptosystems," *Proceedings of the 25th IEEE Symposium on the Foundations of Computer Science*, 1984, pp. 495–503.
662. GSA Federal Standard 1026, "Telecommunications: General Security Requirements for Equipment Using the Data Encryption Standard," General Services Administration, Apr 1982.
663. GSA Federal Standard 1027, "Telecommunications: Interoperability and Security Requirements for Use of the Data Encryption Standard in the Physical and Data Link Layers of Data Communications," General Services Administration, Jan 1983.
664. GSA Federal Standard 1028, "Interoperability and Security Requirements for Use of the Data Encryption Standard with CCITT Group 3 Facsimile Equipment," General Services Administration, Apr 1985.
665. P. Guam, "Cellular Automaton Public Key Cryptosystems," *Complex Systems*, v. 1, 1987, pp. 51–56.
666. H. Guan, "An Analysis of the Finite Automata Public Key Algorithm," *CHINACRYPT '94*, Xidian, China, 11–15 Nov 1994, pp. 120–126. (In Chinese.)
667. G. Guanella, "Means for and Method for Secret Signalling," U.S. Patent #2,405,500, 6 Aug 1946.
668. M. Gude, "Concept for a High-Performance Random Number Generator Based on Physical Random Phenomena," *Frequenz*, v. 39, 1985, pp. 187–190.
669. M. Gude, "Ein quasi-idealer Gleichverteilungsgenerator basierend auf physikalischen Zufallsphänomenen," Ph.D. dissertation, Aachen University of Technology, 1987. (In German.)
670. L.C. Guillou and J.-J. Quisquater, "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 123–128.
671. L.C. Guillou and J.-J. Quisquater, "A 'Paradoxical' Identity-Based Signature Scheme Resulting from Zero-Knowledge," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 216–231.
672. L.C. Guillou, M. Ugon, and J.-J. Quisquater, "The Smart Card: A Standardized

Security Device Dedicated to Public Cryptology," *Contemporary Cryptology: The Science of Information Integrity*, G. Simmons, ed., IEEE Press, 1992, pp. 561–613.

673. C.G. Günther, "Alternating Step Generators Controlled by de Bruijn Sequences," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 5–14.
674. C.G. Günther, "An Identity-based Key-exchange Protocol," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 29–37.
675. H. Gustafson, E. Dawson, and B. Caelli, "Comparison of Block Ciphers," *Advances in Cryptology—AUSCRYPT '90 Proceedings*, Springer-Verlag, 1990, pp. 208–220.
676. P. Gutmann, personal communication, 1993.
677. H. Gutowitz, "A Cellular Automaton Cryptosystem: Specification and Call for Attack," unpublished manuscript, Aug 1992.
678. H. Gutowitz, "Method and Apparatus for Encryption, Decryption, and Authentication Using Dynamical Systems," U.S. Patent #5,365,589, 15 Nov 1994.
679. H. Gutowitz, "Cryptography with Dynamical Systems," *Cellular Automata and Cooperative Phenomenon*, Kluwer Academic Press, 1993.
680. R.K. Guy, "How to Factor a Number," *Fifth Manitoba Conference on Numeral Mathematics Congressus Numerantium*, v. 16, 1976, pp. 49–89.
681. R.K. Guy, *Unsolved Problems in Number Theory*, Springer-Verlag, 1981.
682. S. Haber and W.S. Stornetta, "How to Time-Stamp a Digital Document," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 437–455.
683. S. Haber and W.S. Stornetta, "How to Time-Stamp a Digital Document," *Journal of Cryptology*, v. 3, n. 2, 1991, pp. 99–112.
684. S. Haber and W.S. Stornetta, "Digital Document Time-Stamping with Catenate Certificate," U.S. Patent #5,136,646, 4 Aug 1992.
685. S. Haber and W.S. Stornetta, "Method for Secure Time-Stamping of Digital Documents," U.S. Patent #5,136,647, 4 Aug 1992.
686. S. Haber and W.S. Stornetta, "Method of Extending the Validity of a Cryptographic Certificate," U.S. Patent #5,373,561, 13 Dec 1994.
687. T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A Secret Key Cryptosystem by Iterating a Chaotic Map," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. E73, n. 7, Jul 1990, pp. 1041–1044.
688. T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A Secret Key Cryptosystem by Iterating a Chaotic Map," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 127–140.
689. S. Hada and H. Tanaka, "An Improvement Scheme of DES against Differential Cryptanalysis," *Proceedings of the 1994 Symposium on Cryptography and Information Security (SCIS 94)*, Lake Biwa, Japan, 27–29 Jan 1994, pp 14A.1–11. (In Japanese.)
690. B.C.W. Hagelin, "The Story of the Hagelin Cryptos," *Cryptologia*, v. 18, n. 3, Jul 1994, pp. 204–242.
691. T. Hansen and G.L. Mullen, "Primitive Polynomials over Finite Fields," *Mathematics of Computation*, v. 59, n. 200, Oct 1992, pp. 639–643.
692. S. Harada and S. Kasahara, "An ID-Based Key Sharing Scheme Without Preliminary Communication," IEICE Japan, Technical Report, ISEC89-38, 1989. (In Japanese.)
693. S. Harari, "A Correlation Cryptographic Scheme," *EUROCODE '90—International Symposium on Coding Theory*, Springer-Verlag, 1991, pp. 180–192.
694. T. Hardjono and J. Seberry, "Authentication via Multi-Service Tickets in the Kuperee Server," *Computer Security—ESORICS 94*, Springer-Verlag, 1994, pp. 144–160.
695. L. Harn and T. Kiesler, "New Scheme for Digital Multisignatures," *Electronics Letters*, v. 25, n. 15, 20 Jul 1989, pp. 1002–1003.
696. L. Harn and T. Kiesler, "Improved Rabin's Scheme with High Efficiency," *Electronics Letters*, v. 25, n. 15, 20 Jul 1989, p. 1016.
697. L. Harn and T. Kiesler, "Two New Efficient Cryptosystems Based on Rabin's Scheme," *Fifth Annual Computer Security Applications Conference*, IEEE Computer Society Press, 1990, pp. 263–270.
698. L. Harn and D.-C. Wang, "Cryptanalysis and Modification of Digital Signature Scheme Based on Error-Correcting Codes," *Electronics Letters*, v. 28, n. 2, 10 Jan 1992, p. 157–159.

699. L. Harn and Y. Xu, "Design of Generalized ElGamal Type Digital Signature Schemes Based on Discrete Logarithm," *Electronics Letters*, v. 30, n. 24, 24 Nov 1994, p. 2025–2026.

700. L. Harn and S. Yang, "Group-Oriented Undeniable Signature Schemes without the Assistance of a Mutually Trusted Party," *Advances in Cryptology—AUSCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 133–142.

701. G. Harper, A. Menezes, and S. Vanstone, "Public-Key Cryptosystems with Very Small Key Lengths," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 163–173.

702. C. Harpes, "Notes on High Order Differential Cryptanalysis of DES," internal report, Signal and Information Processing Laboratory, Swiss Federal Institute of Technology, Aug 1993.

703. G.W. Hart, "To Decode Short Cryptograms," *Communications of the ACM*, v. 37, n. 9, Sep 1994, pp. 102–108.

704. J. Hastad, "On Using RSA with Low Exponent in a Public Key Network," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 403–408.

705. J. Hastad and A. Shamir, "The Cryptographic Security of Truncated Linearly Related Variables," *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, 1985, pp. 356–362.

706. R.C. Hauser and E.S. Lee, "Verification and Modelling of Authentication Protocols," *ESORICS 92, Proceedings of the Second European Symposium on Research in Computer Security*, Springer-Verlag, 1992, pp. 131–154.

707. B. Hayes, "Anonymous One-Time Signatures and Flexible Untraceable Electronic Cash," *Advances in Cryptology—AUSCRYPT '90 Proceedings*, Springer-Verlag, 1990, pp. 294–305.

708. D.K. He, "LUC Public Key Cryptosystem and its Properties," *CHINACRYPT '94*, Xidian, China, 11–15 Nov 1994, pp. 60–69. (In Chinese.)

709. J. He and T. Kiesler, "Enhancing the Security of ElGamal's Signature Scheme," *IEE Proceedings on Computers and Digital Techniques*, v. 141, n. 3, 1994, pp. 193–195.

710. E.H. Hebern, "Electronic Coding Machine," U.S. Patent #1,510,441, 30 Sep 1924.

711. N. Heintze and J.D. Tygar, "A Model for Secure Protocols and their Compositions," *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, 1994, pp. 2–13.

712. M.E. Hellman, "An Extension of the Shannon Theory Approach to Cryptography," *IEEE Transactions on Information Theory*, v. IT-23, n. 3, May 1977, pp. 289–294.

713. M.E. Hellman, "The Mathematics of Public-Key Cryptography," *Scientific American*, v. 241, n. 8, Aug 1979, pp. 146–157.

714. M.E. Hellman, "DES Will Be Totally Insecure within Ten Years," *IEEE Spectrum*, v. 16, n. 7, Jul 1979, pp. 32–39.

715. M.E. Hellman, "On DES-Based Synchronous Encryption," Dept. of Electrical Engineering, Stanford University, 1980.

716. M.E. Hellman, "A Cryptanalytic Time-Memory Trade Off," *IEEE Transactions on Information Theory*, v. 26, n. 4, Jul 1980, pp. 401–406.

717. M.E. Hellman, "Another Cryptanalytic Attack on 'A Cryptosystem for Multiple Communications'," *Information Processing Letters*, v. 12, 1981, pp. 182–183.

718. M.E. Hellman, W. Diffie, and R.C. Merkle, "Cryptographic Apparatus and Method," U.S. Patent #4,200,770, 29 Apr 1980.

719. M.E. Hellman, W. Diffie, and R.C. Merkle, "Cryptographic Apparatus and Method," Canada Patent #1,121,480, 6 Apr 1982.

720. M.E. Hellman and R.C. Merkle, "Public Key Cryptographic Apparatus and Method," U.S. Patent #4,218,582, 19 Aug 1980.

721. M.E. Hellman, R. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Pohlig, and P. Schweitzer, "Results of an Initial Attempt to Cryptanalyze the NBS Data Encryption Standard," Technical Report SEL 76-042, Information Systems Lab, Department of Electrical Engineering, Stanford University, 1976.

722. M.E. Hellman and S.C. Pohlig, "Exponentiation Cryptographic Apparatus and Method," U.S. Patent #4,424,414, 3 Jan 1984.

723. M.E. Hellman and J.M. Reyneri, "Distribution of Drainage in the DES," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 129–131.

724. F. Hendessi and M.R. Aref, "A Successful Attack Against the DES," *Third Canadian*

Workshop on Information Theory and Applications, Springer-Verlag, 1994, pp. 78–90.

725. T. Herlestam, "Critical Remarks on Some Public-Key Cryptosystems," *BIT*, v. 18, 1978, pp. 493–496.
726. T. Herlestam, "On Functions of Linear Shift Register Sequences", *Advances in Cryptology—EUROCRYPT '85*, Springer-Verlag, 1986, pp. 119–129.
727. T. Herlestam and R. Johannesson, "On Computing Logarithms over GF(2^p)," *BIT*, v. 21, 1981, pp. 326–334.
728. H.M. Heys and S.E. Tavares, "On the Security of the CAST Encryption Algorithm," *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, Halifax, Nova Scotia, Sep 1994, pp. 332–335.
729. H.M. Heys and S.E. Tavares, "The Design of Substitution-Permutation Networks Resistant to Differential and Linear Cryptanalysis," *Proceedings of the 2nd Annual ACM Conference on Computer and Communications Security*, ACM Press, 1994, pp. 148–155.
730. E. Heyst and T.P. Pederson, "How to Make Fail-Stop Signatures," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 366–377.
731. E. Heyst, T.P. Pederson, and B. Pfitzmann, "New Construction of Fail-Stop Signatures and Lower Bounds," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, pp. 15–30.
732. L.S. Hill, "Cryptography in an Algebraic Alphabet," *American Mathematical Monthly*, v. 36, Jun–Jul 1929, pp. 306–312.
733. P.J.M. Hin, "Channel-Error-Correcting Privacy Cryptosystems," Ph.D. dissertation, Delft University of Technology, 1986. (In Dutch.)
734. R. Hirschfeld, "Making Electronic Refunds Safer," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, pp. 106–112.
735. A. Hodges, *Alan Turing: The Enigma of Intelligence*, Simon and Schuster, 1983.
736. W. Hohl, X. Lai, T. Meier, and C. Waldvogel, "Security of Iterated Hash Functions Based on Block Ciphers," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 379–390.
737. F. Hoornaert, M. Decroos, J. Vandewalle, and R. Govaerts, "Fast RSA-Hardware: Dream or Reality?" *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 257–264.
738. F. Hoornaert, J. Goubert, and Y. Desmedt, "Efficient Hardware Implementation of the DES," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 147–173.
739. E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, Rockville, MD: Computer Science Press, 1978.
740. P. Horster, H. Petersen, and M. Michels, "Meta-ElGamal Signature Schemes," *Proceedings of the 2nd Annual ACM Conference on Computer and Communications Security*, ACM Press, 1994, pp. 96–107.
741. P. Horster, H. Petersen, and M. Michels, "Meta Message Recovery and Meta Blind Signature Schemes Based on the Discrete Logarithm Problem and their Applications," *Advances in Cryptology—ASIACRYPT '94 Proceedings*, Springer-Verlag, 1995, pp. 224–237.
742. L.K. Hua, *Introduction to Number Theory*, Springer-Verlag, 1982.
743. K. Huber, "Specialized Attack on Chor-Rivest Public Key Cryptosystem," *Electronics Letters*, v. 27, n. 23, 7 Nov 1991, pp. 2130–2131.
744. E. Hughes, "A Cypherpunk's Manifesto," 9 Mar 1993.
745. E. Hughes, "An Encrypted Key Transmission Protocol," presented at the rump session of CRYPTO '94, Aug 1994.
746. H. Hule and W.B. Müller, "On the RSA-Cryptosystem with Wrong Keys," *Contributions to General Algebra 6*, Vienna: Verlag Hölder-Pichler-Tempsky, 1988, pp. 103–109.
747. H.A. Hussain, J.W.A. Sada, and S.M. Kalipha, "New Multistage Knapsack Public-Key Cryptosystem," *International Journal of Systems Science*, v. 22, n. 11, Nov 1991, pp. 2313–2320.
748. T. Hwang, "Attacks on Okamoto and Tanaka's One-Way ID-Based Key Distribution System," *Information Processing Letters*, v. 43, n. 2, Aug 1992, pp. 83–86.
749. T. Hwang and T.R.N. Rao, "Secret Error-Correcting Codes (SECC)," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 540–563.
750. C. I'Anson and C. Mitchell, "Security Defects in CCITT Recommendation

X.509—the Directory Authentication Framework," *Computer Communications Review*, v. 20, n. 2, Apr 1990, pp. 30–34.

751. IBM, "Common Cryptographic Architecture: Cryptographic Application Programming Interface Reference," SC40-1675-1, IBM Corp., Nov 1990.
752. IBM, "Common Cryptographic Architecture: Cryptographic Application Programming Interface Reference—Public Key Algorithm," IBM Corp., Mar 1993.
753. R. Impagliazzo and M. Yung, "Direct Minimum-Knowledge Computations," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 40–51.
754. I. Ingemarsson, "A New Algorithm for the Solution of the Knapsack Problem," *Lecture Notes in Computer Science 149; Cryptography: Proceedings of the Workshop on Cryptography*, Springer-Verlag, 1983, pp. 309–315.
755. I. Ingemarsson, "Delay Estimation for Truly Random Binary Sequences or How to Measure the Length of Rip van Winkle's Sleep," *Communications and Cryptography: Two Sides of One Tapestry*, R.E. Blahut et al., eds., Kluwer Adademic Publishers, 1994, pp. 179–186.
756. I. Ingemarsson and G.J. Simmons, "A Protocol to Set Up Shared Secret Schemes without the Assistance of a Mutually Trusted Party," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 266–282.
757. I. Ingemarsson, D.T. Tang, and C.K. Wong, "A Conference Key Distribution System," *IEEE Transactions on Information Theory*, v. IT-28, n. 5, Sep 1982, pp. 714–720.
758. ISO DIS 8730, "Banking—Requirements for Message Authentication (Wholesale)," Association for Payment Clearing Services, London, Jul 1987.
759. ISO DIS 8731-1, "Banking—Approved Algorithms for Message Authentication—Part 1: DEA," Association for Payment Clearing Services, London, 1987.
760. ISO DIS 8731-2, "Banking—Approved Algorithms for Message Authentication—Part 2: Message Authenticator Algorithm," Association for Payment Clearing Services, London, 1987.
761. ISO DIS 8732, "Banking—Key Management (Wholesale)," Association for Payment Clearing Services, London, Dec 1987.
762. ISO/IEC 9796, "Information Technology—Security Techniques—Digital Signature Scheme Giving Message Recovery," International Organization for Standardization, Jul 1991.
763. ISO/IEC 9797, "Data Cryptographic Techniques—Data Integrity Mechanism Using a Cryptographic Check Function Employing a Block Cipher Algorithm," International Organization for Standardization, 1989.
764. ISO DIS 10118 DRAFT, "Information Technology—Security Techniques—Hash Functions," International Organization for Standardization, 1989.
765. ISO DIS 10118 DRAFT, "Information Technology—Security Techniques—Hash Functions," International Organization for Standardization, April 1991.
766. ISO N98, "Hash Functions Using a Pseudo Random Algorithm," working document, ISO-IEC/JTC1/SC27/WG2, International Organization for Standardization, 1992.
767. ISO N179, "AR Fingerprint Function," working document, ISO-IEC/JTC1/SC27/WG2, International Organization for Standardization, 1992.
768. ISO/IEC 10118, "Information Technology—Security Techniques—Hash Functions—Part 1: General and Part 2: Hash-Functions Using an *n*-Bit Block Cipher Algorithm," International Organization for Standardization, 1993.
769. K. Ito, S. Kondo, and Y. Mitsuoka, "SXAL8/MBAL Algorithm," Technical Report, ISEC93-68, IEICE Japan, 1993. (In Japanese.)
770. K.R. Iversen, "The Application of Cryptographic Zero-Knowledge Techniques in Computerized Secret Ballot Election Schemes," Ph.D. dissertation, IDT-report 1991:3, Norwegian Institute of Technology, Feb 1991.
771. K.R. Iversen, "A Cryptographic Scheme for Computerized General Elections," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 405–419.
772. K. Iwamura, T. Matsumoto, and H. Imai, "An Implementation Method for RSA Cryptosystem with Parallel Processing," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. J75-A, n. 8, Aug 1992, pp. 1301–1311.

773. W.J. Jaburek, "A Generalization of ElGamal's Public Key Cryptosystem," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, 1990, Springer-Verlag, pp. 23–28.

774. N.S. James, R. Lidi, and H. Niederreiter, "Breaking the Cade Cipher," *Advances in Cryptology—CRYPTO '86 Proceedings*, 1987, Springer-Verlag, pp. 60–63.

775. C.J.A. Jansen, "On the Key Storage Requirements for Secure Terminals," *Computers and Security*, v. 5, n. 2, Jun 1986, pp. 145–149.

776. C.J.A. Jansen, "Investigations on Nonlinear Streamcipher Systems: Construction and Evaluation Methods," Ph.D. dissertation, Technical University of Delft, 1989.

777. C.J.A. Jansen and D.E. Boekee, "Modes of Blockcipher Algorithms and their Protection against Active Eavesdropping," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 281–286.

778. S.M. Jennings, "A Special Class of Binary Sequences," Ph.D. dissertation, University of London, 1980.

779. S.M. Jennings, "Multiplexed Sequences: Some Properties of the Minimum Polynomial," *Lecture Notes in Computer Science 149; Cryptography: Proceedings of the Workshop on Cryptography*, Springer-Verlag, 1983, pp. 189–206.

780. S.M. Jennings, "Autocorrelation Function of the Multiplexed Sequence," *IEE Proceedings*, v. 131, n. 2, Apr 1984, pp. 169–172.

781. T. Jin, "Care and Feeding of Your Three-Headed Dog," Document Number IAG-90-011, Hewlett-Packard, May 1990.

782. T. Jin, "Living with Your Three-Headed Dog," Document Number IAG-90-012, Hewlett-Packard, May 1990.

783. A. Jiwa, J. Seberry, and Y. Zheng, "Beacon Based Authentication," *Computer Security—ESORICS 94*, Springer-Verlag, 1994, pp. 125–141.

784. D.B. Johnson, G.M. Dolan, M.J. Kelly, A.V. Le, and S.M. Matyas, "Common Cryptographic Architecture Cryptographic Application Programming Interface," *IBM Systems Journal*, v. 30, n. 2, 1991, pp. 130–150.

785. D.B. Johnson, S.M. Matyas, A.V. Le, and J.D. Wilkins, "Design of the Commercial Data Masking Facility Data Privacy Algorithm," *1st ACM Conference on Computer and Communications Security*, ACM Press, 1993, pp. 93–96.

786. J.P. Jordan, "A Variant of a Public-Key Cryptosystem Based on Goppa Codes," *Sigact News*, v. 15, n. 1, 1983, pp. 61–66.

787. A. Joux and L. Granboulan, "A Practical Attack Against Knapsack Based Hash Functions," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.

788. A. Joux and J. Stern, "Cryptanalysis of Another Knapsack Cryptosystem," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 470–476.

789. R.R. Jueneman, "Analysis of Certain Aspects of Output-Feedback Mode," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 99–127.

790. R.R. Jueneman, "Electronic Document Authentication," *IEEE Network Magazine*, v. 1, n. 2, Apr 1978, pp. 17–23.

791. R.R. Jueneman, "A High Speed Manipulation Detection Code," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 327–346.

792. R.R. Jueneman, S.M. Matyas, and C.H. Meyer, "Message Authentication with Manipulation Detection Codes," *Proceedings of the 1983 IEEE Computer Society Symposium on Research in Security and Privacy*, 1983, pp. 733–54.

793. R.R. Jueneman, S.M. Matyas, and C.H. Meyer, "Message Authentication," *IEEE Communications Magazine*, v. 23, n. 9, Sep 1985, pp. 29–40.

794. D. Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Macmillan Publishing Co., 1967.

795. D. Kahn, *Kahn on Codes*, New York: Macmillan Publishing Co., 1983.

796. D. Kahn, *Seizing the Enigma*, Boston: Houghton Mifflin Co., 1991.

797. P. Kaijser, T. Parker, and D. Pinkas, "SESAME: The Solution to Security for Open Distributed Systems," *Journal of Computer Communications*, v. 17, n. 4, Jul 1994, pp. 501–518.

798. R. Kailar and V.D. Gilgor, "On Belief Evolution in Authentication Protocols," *Proceedings of the Computer Security Foundations Workshop IV*, IEEE Computer Society Press, 1991, pp. 102–116.

799. B.S. Kaliski, "A Pseudo Random Bit Generator Based on Elliptic Logarithms," Master's thesis, Massachusetts Institute of Technology, 1987.
800. B.S. Kaliski, letter to NIST regarding DSS, 4 Nov 1991.
801. B.S. Kaliski, "The MD2 Message Digest Algorithm," RFC 1319, Apr 1992.
802. B.S. Kaliski, "Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certificates and Related Services," RFC 1424, Feb 1993.
803. B.S. Kaliski, "An Overview of the PKCS Standards," RSA Laboratories, Nov 1993.
804. B.S. Kaliski, "A Survey of Encryption Standards, *IEEE Micro*, v. 13, n. 6, Dec 1993, pp. 74–81.
805. B.S. Kaliski, personal communication, 1993.
806. B.S. Kaliski, "On the Security and Performance of Several Triple-DES Modes," RSA Laboratories, draft manuscript, Jan 1994.
807. B.S. Kaliski, R.L. Rivest, and A.T. Sherman, "Is the Data Encryption Standard a Group?", *Advances in Cryptology—EUROCRYPT '85*, Springer-Verlag, 1986, pp. 81–95.
808. B.S. Kaliski, R.L. Rivest, and A.T. Sherman, "Is the Data Encryption Standard a Pure Cipher? (Results of More Cycling Experiments in DES)," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 212–226.
809. B.S. Kaliski, R.L. Rivest, and A.T. Sherman, "Is the Data Encryption Standard a Group? (Results of Cycling Experiments on DES)," *Journal of Cryptology*, v. 1, n. 1, 1988, pp. 3–36.
810. B.S. Kaliski and M.J.B. Robshaw, "Fast Block Cipher Proposal," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 33–40.
811. B.S. Kaliski and M.J.B. Robshaw, "Linear Cryptanalysis Using Multiple Approximations," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994, pp. 26–39.
812. B.S. Kaliski and M.J.B. Robshaw, "Linear Cryptanalysis Using Multiple Approximations and FEAL," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
813. R.G. Kammer, statement before the U.S. government Subcommittee on Telecommunications and Finance, Committee on Energy and Commerce, 29 Apr 1993.
814. T. Kaneko, K. Koyama, and R. Terada, "Dynamic Swapping Schemes and Differential Cryptanalysis, *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 292–301.
815. T. Kaneko, K. Koyama, and R. Terada, "Dynamic Swapping Schemes and Differential Cryptanalysis," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. E77-A, n. 8, Aug 1994, pp. 1328–1336.
816. T. Kaneko and H. Miyano, "A Study on the Strength Evaluation of Randomized DES-Like Cryptosystems against Chosen Plaintext Attacks," *Proceedings of the 1993 Symposium on Cryptography and Information Security (SCIS 93)*, Shuzenji, Japan, 28–30 Jan 1993, pp. 15C.1–10.
817. J. Kari, "A Cryptosystem Based on Propositional Logic," *Machines, Languages, and Complexity: 5th International Meeting of Young Computer Scientists, Selected Contributions*, Springer-Verlag, 1989, pp. 210–219.
818. E.D. Karnin, J.W. Greene, and M.E. Hellman, "On Sharing Secret Systems," *IEEE Transactions on Information Theory*, v. IT-29, 1983, pp. 35–41.
819. F.W. Kasiski, *Die Geheimschriften und die Dechiffrir-kunst*, E.S. Miller und Sohn, 1863. (In German.)
820. A. Kehne, J. Schonwalder, and H. Langendorfer, "A Nonce-Based Protocol for Multiple Authentications," *Operating Systems Review*, v. 26, n. 4, Oct 1992, pp. 84–89.
821. J. Kelsey, personal communication, 1994.
822. R. Kemmerer, "Analyzing Encryption Protocols Using Formal Verification Techniques," *IEEE Journal on Selected Areas in Communications*, v. 7, n. 4, May 1989, pp. 448–457.
823. R. Kemmerer, C.A. Meadows, and J. Millen, "Three Systems for Cryptographic Protocol Analysis," *Journal of Cryptology*, v. 7, n. 2, 1994, pp. 79–130.
824. S.T. Kent, "Encryption-Based Protection Protocols for Interactive User-Computer Communications," MIT/LCS/TR-162,

MIT Laboratory for Computer Science, May 1976.

825. S.T. Kent, "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management," RFC 1422, Feb 1993.
826. S.T. Kent, "Understanding the Internet Certification System," *Proceedings of INET '93*, The Internet Society, 1993, pp. BAB1-BAB10.
827. S.T. Kent and J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management," RFC 1114, Aug 1989.
828. V. Kessler and G. Wedel, "AUTOLOG—An Advanced Logic of Authentication," *Proceedings of the Computer Security Foundations Workshop VII*, IEEE Computer Society Press, 1994, pp. 90–99.
829. E.L. Key, "An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators," *IEEE Transactions on Information Theory*, v. IT-22, n. 6, Nov 1976, pp. 732–736.
830. T. Kiesler and L. Harn, "RSA Blocking and Multisignature Schemes with No Bit Expansion," *Electronics Letters*, v. 26, n. 18, 30 Aug 1990, pp. 1490–1491.
831. J. Kilian, *Uses of Randomness in Algorithms and Protocols*, MIT Press, 1990.
832. J. Kilian, "Achieving Zero-Knowledge Robustly," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 313–325.
833. J. Kilian and T. Leighton, "Failsafe Key Escrow," MIT/LCS/TR-636, MIT Laboratory for Computer Science, Aug 1994.
834. K. Kim, "Construction of DES-Like S-Boxes Based on Boolean Functions Satisfying the SAC," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 59–72.
835. K. Kim, S. Lee, and S. Park, "Necessary Conditions to Strengthen DES S-Boxes Against Linear Cryptanalysis," *Proceedings of the 1994 Symposium on Cryptography and Information Security (SCIS 94)*, Lake Biwa, Japan, 27–29 Jan 1994, pp. 15D.1–9.
836. K. Kim, S. Lee, and S. Park, "How to Strengthen DES against Differential Attack," unpublished manuscript, 1994.
837. K. Kim, S. Lee, S. Park, and D. Lee, "DES Can Be Immune to Differential Cryptanalysis," *Workshop on Selected Areas in Cryptography—Workshop Record*, Kingston, Ontario, 5–6 May 1994, pp. 70–81.
838. K. Kim, S. Park, and S. Lee, "How to Strengthen DES against Two Robust Attacks," *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography*, Inuyama, Japan, 24–27 Jan 1995, 173–182.
839. K. Kim, S. Park, and S. Lee, "Reconstruction of s^2DES S-Boxes and their Immunity to Differential Cryptanalysis," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 282–291.
840. S. Kim and B.S. Um, "A Multipurpose Membership Proof System Based on Discrete Logarithm," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 177–183.
841. P. Kinnucan, "Data Encryption Gurus: Tuchman and Meyer," *Cryptologia*, v. 2, n. 4, Oct 1978.
842. A. Klapper, "The Vulnerability of Geometric Sequences Based on Fields of Odd Characteristic," *Journal of Cryptology*, v. 7, n. 1, 1994, pp. 33–52.
843. A. Klapper, "Feedback with Carry Shift Registers over Finite Fields," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.
844. A. Klapper and M. Goresky, "2-adic Shift Registers," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 174–178.
845. A. Klapper and M. Goresky, "2-adic Shift Registers," Technical Report #239-93, Department of Computer Science, University of Kentucky, 19 Apr 1994.
846. A. Klapper and M. Goresky, "Large Period Nearly de Bruijn FCSR Sequences," *Advances in Cryptology—EUROCRYPT '95 Proceedings*, Springer-Verlag, 1995, pp. 263–273.
847. D.V. Klein, " 'Foiling the Cracker': A Survey of, and Implications to, Password Security," *Proceedings of the USENIX UNIX Security Workshop*, Aug 1990, pp. 5–14.
848. D.V. Klein, personal communication, 1994.
849. C.S. Kline and G.J. Popek, "Public Key vs. Conventional Key Cryptosystems," *Pro-*

ceedings of AFIPS National Computer Conference, pp. 831–837.

850. H.-J. Knobloch, "A Smart Card Implementation of the Fiat-Shamir Identification Scheme," *Advances in Cryptology—EUROCRPYT '88 Proceedings,* Springer-Verlag, 1988, pp. 87–95.
851. T. Knoph, J. Frößl, W. Beller, and T. Giesler, "A Hardware Implementation of a Modified DES Algorithm," *Microprocesssing and Microprogramming,* v. 30, 1990, pp. 59–66.
852. L.R. Knudsen, "Cryptanalysis of LOKI," *Advances in Cryptology—ASIACRYPT '91 Proceedings,* Springer-Verlag, 1993, pp. 22–35.
853. L.R. Knudsen, "Cryptanalysis of LOKI," *Cryptography and Coding III,* M.J. Ganley, ed., Oxford: Clarendon Press, 1993, pp. 223–236.
854. L.R. Knudsen, "Cryptanalysis of LOKI91," *Advances in Cryptology—AUSCRYPT '92 Proceedings,* Springer-Verlag, 1993, pp. 196–208.
855. L.R. Knudsen, "Iterative Characteristics of DES and s^2DES," *Advances in Cryptology—CRYPTO '92,* Springer-Verlag, 1993, pp. 497–511.
856. L.R. Knudsen, "An Analysis of Kim, Park, and Lee's DES-Like S-Boxes," unpublished manuscript, 1993.
857. L.R. Knudsen, "Practically Secure Feistel Ciphers," *Fast Software Encryption, Cambridge Security Workshop Proceedings,* Springer-Verlag, 1994, pp. 211–221.
858. L.R. Knudsen, "Block Ciphers—Analysis, Design, Applications," Ph.D. dissertation, Aarhus University, Nov 1994.
859. L.R. Knudsen, personal communication, 1994.
860. L.R. Knudsen, "Applications of Higher Order Differentials and Partial Differentials," *K.U. Leuven Workshop on Cryptographic Algorithms,* Springer-Verlag, 1995, to appear.
861. L.R. Knudsen and X. Lai, "New Attacks on All Double Block Length Hash Functions of Hash Rate 1, Including the Parallel-DM," *Advances in Cryptology—EUROCRYPT '94 Proceedings,* Springer-Verlag, 1995, to appear.
862. L.R. Knudsen, "A Weakness in SAFER K-64," *Advances in Cryptology–CRYPTO '95 Proceedings,* Springer-Verlag, 1995, to appear.
863. D. Knuth, *The Art of Computer Programming: Volume 2, Seminumerical Algorithms,* 2nd edition, Addison-Wesley, 1981.
864. D. Knuth, "Deciphering a Linear Congruential Encryption," *IEEE Transactions on Information Theory,* v. IT-31, n. 1, Jan 1985, pp. 49–52.
865. K. Kobayashi and L. Aoki, "On Linear Cryptanalysis of MBAL," *Proceedings of the 1995 Symposium on Cryptography and Information Security (SCIS 95),* Inuyama, Japan, 24–27 Jan 1995, pp. A4.2.1–9.
866. K. Kobayashi, K. Tamura, and Y. Nemoto, "Two-dimensional Modified Rabin Cryptosystem," *Transactions of the Institute of Electronics, Information, and Communication Engineers,* v. J72-D, n. 5, May 1989, pp. 850–851. (In Japanese.)
867. N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation,* v. 48, n. 177, 1987, pp. 203–209.
868. N. Koblitz, "A Family of Jacobians Suitable for Discrete Log Cryptosystems," *Advances in Cryptology—CRYPTO '88 Proceedings,* Springer-Verlag, 1990, pp. 94–99.
869. N. Koblitz, "Constructing Elliptic Curve Cryptosystems in Characteristic 2," *Advances in Cryptology—CRYPTO '90 Proceedings,* Springer-Verlag, 1991, pp. 156–167.
870. N. Koblitz, "Hyperelliptic Cryptosystems," *Journal of Cryptology,* v. 1, n. 3, 1989, pp. 129–150.
871. N. Koblitz, "CM-Curves with Good Cryptographic Properties," *Advances in Cryptology—CRYPTO '91 Proceedings,* Springer-Verlag, 1992, pp. 279–287.
872. Ç.K. Koç, "High-Speed RSA Implementation," Version 2.0, RSA Laboratories, Nov 1994.
873. M.J. Kochanski, "Remarks on Lu and Lee's Proposals," *Cryptologia,* v. 4, n. 4, 1980, pp. 204–207.
874. M.J. Kochanski, "Developing an RSA Chip," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 350–357.
875. J.T. Kohl, "The Use of Encryption in Kerberos for Network Authentication," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 35–43.

876. J.T. Kohl, "The Evolution of the Kerberos Authentication Service," *EurOpen Conference Proceedings*, May 1991, pp. 295–313.
877. J.T. Kohl and B.C. Neuman, "The Kerberos Network Authentication Service," RFC 1510, Sep 1993.
878. J.T. Kohl, B.C. Neuman, and T. Ts'o, "The Evolution of the Kerberos Authentication System," *Distributed Open Systems*, IEEE Computer Society Press, 1994, pp. 78–94.
879. Kohnfelder, "Toward a Practical Public Key Cryptosystem," Bachelor's thesis, MIT Department of Electrical Engineering, May 1978.
880. A.G. Konheim, *Cryptography: A Primer*, New York: John Wiley & Sons, 1981.
881. A.G. Konheim, M.H. Mack, R.K. McNeill, B. Tuckerman, and G. Waldbaum, "The IPS Cryptographic Programs," *IBM Systems Journal*, v. 19, n. 2, 1980, pp. 253–283.
882. V.I. Korzhik and A.I. Turkin, "Cryptanalysis of McEliece's Public-Key Cryptosystem," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 68–70.
883. S.C. Kothari, "Generalized Linear Threshold Scheme," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 231–241.
884. J. Kowalchuk, B.P. Schanning, and S. Powers, "Communication Privacy: Integration of Public and Secret Key Cryptography," *Proceedings of the National Telecommunication Conference*, IEEE Press, 1980, pp. 49.1.1–49.1.5.
885. K. Koyama, "A Master Key for the RSA Public-Key Cryptosystem," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. J65-D, n. 2, Feb 1982, pp. 163–170.
886. K. Koyama, "A Cryptosystem Using the Master Key for Multi-Address Communications," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. J65-D, n. 9, Sep 1982, pp. 1151–1158.
887. K. Koyama, "Demonstrating Membership of a Group Using the Shizuya-Koyama-Itoh (SKI) Protocol," *Proceedings of the 1989 Symposium on Cryptography and Information Security (SCIS 89)*, Gotenba, Japan, 1989.
888. K. Koyama, "Direct Demonstration of the Power to Break Public-Key Cryptosystems," *Advances in Cryptology—AUSCRYPT '90 Proceedings*, Springer-Verlag, 1990, pp. 14–21.
889. K. Koyama, "Security and Unique Decipherability of Two-dimensional Public Key Cryptosystems," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. E73, n. 7, Jul 1990, pp. 1057–1067.
890. K. Koyama, U.M. Maurer, T. Okamoto, and S.A. Vanstone, "New Public-Key Schemes Based on Elliptic Curves over the Ring Z_n," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 252–266.
891. K. Koyama and K. Ohta, "Identity-based Conference Key Distribution System," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 175–184.
892. K. Koyama and T. Okamoto, "Elliptic Curve Cryptosystems and Their Applications," *IEICE Transactions on Information and Systems*, v. E75-D, n. 1, Jan 1992, pp. 50–57.
893. K. Koyama and R. Terada, "How to Strengthen DES-Like Cryptosystems against Differential Cryptanalysis," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. E76-A, n. 1, Jan 1993, pp. 63–69.
894. K. Koyama and R. Terada, "Probabilistic Swapping Schemes to Strengthen DES against Differential Cryptanalysis," *Proceedings of the 1993 Symposium on Cryptography and Information Security (SCIS 93)*, Shuzenji, Japan, 28–30 Jan 1993, pp. 15D.1–12.
895. K. Koyama and Y. Tsuruoka, "Speeding up Elliptic Cryptosystems Using a Singled Binary Window Method," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, pp. 345–357.
896. E. Kranakis, *Primality and Cryptography*, Wiler-Teubner Series in Computer Science, 1986.
897. D. Kravitz, "Digital Signature Algorithm," U.S. Patent #5,231,668, 27 Jul 1993.
898. D. Kravitz and I. Reed, "Extension of RSA Cryptostructure: A Galois Approach," *Electronics Letters*, v. 18, n. 6, 18 Mar 1982, pp. 255–256.

899. H. Krawczyk, "How to Predict Congruential Generators," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 138–153.
900. H. Krawczyk, "How to Predict Congruential Generators," *Journal of Algorithms*, v. 13, n. 4, Dec 1992, pp. 527–545.
901. H. Krawczyk, "The Shrinking Generator: Some Practical Considerations," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 45–46.
902. G.J. Kühn, "Algorithms for Self-Synchronizing Ciphers," *Proceedings of COMSIG 88*, 1988.
903. G.J. Kühn, F. Bruwer, and W. Smit, "'n Vinnige Veeldoelige Enkripsievlokkie," *Proceedings of Infosec 90*, 1990. (In Afrikaans.)
904. S. Kullback, *Statistical Methods in Cryptanalysis*, U.S. Government Printing Office, 1935. Reprinted by Aegean Park Press, 1976.
905. P.V. Kumar, R.A. Scholtz, and L.R. Welch, "Generalized Bent Functions and their Properties," *Journal of Combinational Theory*, Series A, v. 40, n. 1, Sep 1985, pp. 90–107.
906. M. Kurosaki, T. Matsumoto, and H. Imai, "Simple Methods for Multipurpose Certification," *Proceedings of the 1989 Symposium on Cryptography and Information Security (SCIS 89)*, Gotenba, Japan, 1989.
907. M. Kurosaki, T. Matsumoto, and H. Imai, "Proving that You Belong to at Least One of the Specified Groups," *Proceedings of the 1990 Symposium on Cryptography and Information Security (SCIS 90)*, Hihondaira, Japan, 1990.
908. K. Kurosawa, "Key Changeable ID-Based Cryptosystem," *Electronics Letters*, v. 25, n. 9, 27 Apr 1989, pp. 577–578.
909. K. Kurosawa, T. Ito, and M. Takeuchi, "Public Key Cryptosystem Using a Reciprocal Number with the Same Intractability as Factoring a Large Number," *Cryptologia*, v. 12, n. 4, Oct 1988, pp. 225–233.
910. K. Kurosawa, C. Park, and K. Sakano, "Group Signer/Verifier Separation Scheme," *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography*, Inuyama, Japan, 24–27 Jan 1995, 134–143.
911. G.C. Kurtz, D. Shanks, and H.C. Williams, "Fast Primality Tests for Numbers Less than $50*10^9$," *Mathematics of Computation*, v. 46, n. 174, Apr 1986, pp. 691–701.
912. K. Kusuda and T. Matsumoto, "Optimization of the Time-Memory Trade-Off Cryptanalysis and Its Application to Block Ciphers," *Proceedings of the 1995 Symposium on Cryptography and Information Security (SCIS 95)*, Inuyama, Japan, 24–27 Jan 1995, pp. A3.2.1–11. (In Japanese.)
913. H. Kuwakado and K. Koyama, "Security of RSA-Type Cryptosystems Over Elliptic Curves against Hastad Attack," *Electronics Letters*, v. 30, n. 22, 27 Oct 1994, pp. 1843–1844.
914. H. Kuwakado and K. Koyama, "A New RSA-Type Cryptosystem over Singular Elliptic Curves," *IMA Conference on Applications of Finite Fields*, Oxford University Press, to appear.
915. H. Kuwakado and K. Koyama, "A New RSA-Type Scheme Based on Singular Cubic Curves," *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography*, Inuyama, Japan, 24–27 Jan 1995, pp. 144–151.
916. M. Kwan, "An Eight Bit Weakness in the LOKI Cryptosystem," technical report, Australian Defense Force Academy, Apr 1991.
917. M. Kwan and J. Pieprzyk, "A General Purpose Technique for Locating Key Scheduling Weakness in DES-Like Cryptosystems," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 237–246.
918. J.B. Lacy, D.P. Mitchell, and W.M. Schell, "CryptoLib: Cryptography in Software," *UNIX Security Symposium IV Proceedings*, USENIX Association, 1993, pp. 1–17.
919. J.C. Lagarias, "Knapsack Public Key Cryptosystems and Diophantine Approximations," *Advances in Cryptology: Proceedings of Crypto 83*, Plenum Press, 1984, pp. 3–23.
920. J.C. Lagarias, "Performance Analysis of Shamir's Attack on the Basic Merkle-Hellman Knapsack Cryptosystem," *Lecture Notes in Computer Science 172; Proceedings of the 11th International Colloquium on Automata, Languages, and Programming (ICALP)*, Springer-Verlag, 1984, pp. 312–323.
921. J.C. Lagarias and A.M. Odlyzko, "Solving Low-Density Subset Sum Problems," *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, 1983, pp. 1–10.

922. J.C. Lagarias and A.M. Odlyzko, "Solving Low-Density Subset Sum Problems," *Journal of the ACM*, v. 32, n. 1, Jan 1985, pp. 229–246.

923. J.C. Lagarias and J. Reeds, "Unique Extrapolation of Polynomial Recurrences," *SIAM Journal on Computing*, v. 17, n. 2, Apr 1988, pp. 342–362.

924. X. Lai, *Detailed Description and a Software Implementation of the IPES Cipher*, unpublished manuscript, 8 Nov 1991.

925. X. Lai, *On the Design and Security of Block Ciphers*, ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992.

926. X. Lai, personal communication, 1993.

927. X. Lai, "Higher Order Derivatives and Differential Cryptanalysis," *Communications and Cryptography: Two Sides of One Tapestry*, R.E. Blahut et al., eds., Kluwer Adademic Publishers, 1994, pp. 227–233.

928. X. Lai and L. Knudsen, "Attacks on Double Block Length Hash Functions," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 157–165.

929. X. Lai and J. Massey, "A Proposal for a New Block Encryption Standard," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 389–404.

930. X. Lai and J. Massey, "Hash Functions Based on Block Ciphers," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1992, pp. 55–70.

931. X. Lai, J. Massey, and S. Murphy, "Markov Ciphers and Differential Cryptanalysis," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 17–38.

932. X. Lai, R.A. Rueppel, and J. Woollven, "A Fast Cryptographic Checksum Algorithm Based on Stream Ciphers," *Advances in Cryptology—AUSCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 339–348.

933. C.S. Laih, J.Y. Lee, C.H. Chen, and L. Harn, "A New Scheme for ID-based Cryptosystems and Signatures," *Journal of the Chinese Institute of Engineers*, v. 15, n. 2, Sep 1992, pp. 605–610.

934. B.A. LaMacchia and A.M. Odlyzko, "Computation of Discrete Logarithms in Prime Fields," *Designs, Codes, and Cryptography*, v. 1, 1991, pp. 46–62.

935. L. Lamport, "Password Identification with Insecure Communications," *Communications of the ACM*, v. 24, n. 11, Nov 1981, pp. 770–772.

936. S. Landau, "Zero-Knowledge and the Department of Defense," *Notices of the American Mathematical Society*, v. 35, n. 1, Jan 1988, pp. 5–12.

937. S. Landau, S. Kent, C. Brooks, S. Charney, D. Denning, W. Diffie, A. Lauck, D. Mikker, P. Neumann, and D. Sobel, "Codes, Keys, and Conflicts: Issues in U.S. Crypto Policy," Report of a Special Panel of the ACM U.S. Public Policy Committee (USACM), Association for Computing Machinery, Jun 1994.

938. S.K. Langford and M.E. Hellman, "Cryptanalysis of DES," presented at 1994 RSA Data Security conference, Redwood Shores, CA, 12–14 Jan 1994.

939. D. Lapidot and A. Shamir, "Publicly Verifiable Non-Interactive Zero-Knowledge Proofs," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 353–365.

940. A.V. Le, S.M. Matyas, D.B. Johnson, and J.D. Wilkins, "A Public-Key Extension to the Common Cryptographic Architecture," *IBM Systems Journal*, v. 32, n. 3, 1993, pp. 461–485.

941. P. L'Ecuyer, "Efficient and Portable Combined Random Number Generators," *Communications of the ACM*, v. 31, n. 6, Jun 1988, pp. 742–749, 774.

942. P. L'Ecuyer, "Random Numbers for Simulation," *Communications of the ACM*, v. 33, n. 10, Oct 1990, pp. 85–97.

943. P.J. Lee and E.F. Brickell, "An Observation on the Security of McEliece's Public-Key Cryptosystem," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 275–280.

944. S. Lee, S. Sung, and K. Kim, "An Efficient Method to Find the Linear Expressions for Linear Cryptanalysis," *Proceedings of the 1995 Korea-Japan Workshop on Information Security and Cryptography*, Inuyama, Japan, 24–26 Jan 1995, pp. 183–190.

945. D.J. Lehmann, "On Primality Tests," *SIAM Journal on Computing*, v. 11, n. 2, May 1982, pp. 374–375.

946. T. Leighton, "Failsafe Key Escrow Systems," Technical Memo 483, MIT Laboratory for Computer Science, Aug 1994.

947. A. Lempel and M. Cohn, "Maximal Families of Bent Sequences," *IEEE Transactions on Information Theory,* v. IT-28, n. 6, Nov 1982, pp. 865–868.
948. A.K. Lenstra, "Factoring Multivariate Polynomials Over Finite Fields," *Journal of Computer System Science,* v. 30, n. 2, Apr 1985, pp. 235–248.
949. A.K. Lenstra, personal communication, 1995.
950. A.K. Lenstra and S. Haber, letter to NIST Regarding DSS, 26 Nov 1991.
951. A.K. Lenstra, H.W. Lenstra Jr., and L. Lovácz, "Factoring Polynomials with Rational Coefficients," *Mathematische Annalen,* v. 261, n. 4, 1982, pp. 515–534.
952. A.K. Lenstra, H.W. Lenstra, Jr., M.S. Manasse, and J.M. Pollard, "The Number Field Sieve," *Proceedings of the 22nd ACM Symposium on the Theory of Computing,* 1990, pp. 574–572.
953. A.K. Lenstra and H.W. Lenstra, Jr., eds., *Lecture Notes in Mathematics 1554: The Development of the Number Field Sieve,* Springer-Verlag, 1993.
954. A.K. Lenstra, H.W. Lenstra, Jr., M.S. Manasse, and J.M. Pollard, "The Factorization of the Ninth Fermat Number," *Mathematics of Computation,* v. 61, n. 203, 1993, pp. 319–349.
955. A.K. Lenstra and M.S. Manasse, "Factoring by Electronic Mail," *Advances in Cryptology—EUROCRYPT '89 Proceedings,* Springer-Verlag, 1990, pp. 355–371.
956. A.K. Lenstra and M.S. Manasse, "Factoring with Two Large Primes," *Advances in Cryptology—EUROCRYPT '90 Proceedings,* Springer-Verlag, 1991, pp. 72–82.
957. H.W. Lenstra Jr. "Elliptic Curves and Number-Theoretic Algorithms," Report 86-19, Mathematisch Instituut, Universiteit van Amsterdam, 1986.
958. H.W. Lenstra Jr. "On the Chor-Rivest Knapsack Cryptosystem," *Journal of Cryptology,* v. 3, n. 3, 1991, pp. 149–155.
959. W.J. LeVeque, *Fundamentals of Number Theory,* Addison-Wesley, 1977.
960. L.A. Levin, "One-Way Functions and Pseudo-Random Generators," *Proceedings of the 17th ACM Symposium on Theory of Computing,* 1985, pp. 363–365.
961. Lexar Corporation, "An Evaluation of the DES," Sep 1976.
962. D.-X. Li, "Cryptanalysis of Public-Key Distribution Systems Based on Dickson Polynomials," *Electronics Letters,* v. 27, n. 3, 1991, pp. 228–229.
963. F.-X. Li, "How to Break Okamoto's Cryptosystems by Continued Fraction Algorithm," *ASIACRYPT '91 Abstracts,* 1991, pp. 285–289.
964. Y.X. Li and X.M. Wang, "A Joint Authentication and Encryption Scheme Based on Algebraic Coding Theory," *Applied Algebra, Algebraic Algorithms and Error Correcting Codes 9,* Springer-Verlag, 1991, pp. 241–245.
965. R. Lidl, G.L. Mullen, and G. Turwald, *Pitman Monographs and Surveys in Pure and Applied Mathematics 65: Dickson Polynomials,* London: Longman Scientific and Technical, 1993.
966. R. Lidl and W.B. Müller, "Permutation Polynomials in RSA-Cryptosystems," *Advances in Cryptology: Proceedings of Crypto 83,* Plenum Press, 1984, pp. 293–301.
967. R. Lidl and W.B. Müller, "Generalizations of the Fibonacci Pseudoprimes Test," *Discrete Mathematics,* v. 92, 1991, pp. 211–220.
968. R. Lidl and W.B. Müller, "Primality Testing with Lucas Functions," *Advances in Cryptology—AUSCRYPT '92 Proceedings,* Springer-Verlag, 1993, pp. 539–542.
969. R. Lidl, W.B. Müller, and A. Oswald, "Some Remarks on Strong Fibonacci Pseudoprimes," *Applicable Algebra in Engineering, Communication and Computing,* v. 1, n. 1, 1990, pp. 59–65.
970. R. Lidl and H. Niederreiter, "Finite Fields," *Encyclopedia of Mathematics and its Applications,* v. 20, Addison-Wesley, 1983.
971. R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications,* London: Cambridge University Press, 1986.
972. K. Lieberherr, "Uniform Complexity and Digital Signatures," *Theoretical Computer Science,* v. 16, n. 1, Oct 1981, pp. 99–110.
973. C.H. Lim and P.J. Lee, "A Practical Electronic Cash System for Smart Cards," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography,* Seoul, Korea, 24–26 Oct 1993, pp. 34–47.
974. C.H. Lim and P.J. Lee, "Security of Interactive DSA Batch Verification," *Electronics Letters,* v. 30, n. 19, 15 Sep 1994, pp. 1592–1593.

975. H.-Y. Lin and L. Harn, "A Generalized Secret Sharing Scheme with Cheater Detection," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 149–158.

976. M.-C. Lin, T.-C. Chang, and H.-L. Fu, "Information Rate of McEliece's Public-key Cryptosystem," *Electronics Letters*, v. 26, n. 1, 4 Jan 1990, pp. 16–18.

977. J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures," RFC 989, Feb 1987.

978. J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures," RFC 1040, Jan 1988.

979. J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures," RFC 1113, Aug 1989.

980. J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part III—Algorithms, Modes, and Identifiers," RFC 1115, Aug 1989.

981. J. Linn, "Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures," RFC 1421, Feb 1993.

982. S. Lloyd, "Counting Binary Functions with Certain Cryptographic Properties," *Journal of Cryptology*, v. 5, n. 2, 1992, pp. 107–131.

983. T.M.A. Lomas, "Collision-Freedom, Considered Harmful, or How to Boot a Computer," *Proceedings of the 1995 Korea-Japan Workshop on Information Security and Cryptography*, Inuyama, Japan, 24–26 Jan 1995, pp. 35–42.

984. T.M.A. Lomas and M. Roe, "Forging a Clipper Message," *Communications of the ACM*, v. 37, n. 12, 1994, p. 12.

985. D.L. Long, "The Security of Bits in the Discrete Logarithm," Ph.D. dissertation, Princeton University, Jan 1984.

986. D.L. Long and A. Wigderson, "How Discrete Is the Discrete Log," *Proceedings of the 15th Annual ACM Syposium on the Theory of Computing*, Apr 1983.

987. D. Longley and S. Rigby, "An Automatic Search for Security Flaws in Key Management Schemes," *Computers and Security*, v. 11, n. 1, Jan 1992. pp. 75–89.

988. S.H. Low, N.F. Maxemchuk, and S. Paul, "Anonymous Credit Cards," *Proceedings of the 2nd Annual ACM Conference on Computer and Communications Security*, ACM Press, 1994, pp. 108–117.

989. J.H. Loxton, D.S.P. Khoo, G.J. Bird, and J. Seberry, "A Cubic RSA Code Equivalent to Factorization," *Journal of Cryptology*, v. 5, n. 2, 1992, pp. 139–150.

990. S.C. Lu and L.N. Lee, "A Simple and Effective Public-Key Cryptosystem," *COMSAT Technical Review*, 1979, pp. 15–24.

991. M. Luby, S. Micali, and C. Rackoff, "How to Simultaneously Exchange a Secret Bit by Flipping a Symmetrically-Biased Coin," *Proceedings of the 24nd Annual Symposium on the Foundations of Computer Science*, 1983, pp. 11–22.

992. M. Luby and C. Rackoff, "How to Construct Pseudo-Random Permutations from Pseudorandom Functions," *SIAM Journal on Computing*, Apr 1988, pp. 373–386.

993. F. Luccio and S. Mazzone, "A Cryptosystem for Multiple Communications," *Information Processing Letters*, v. 10, 1980, pp. 180–183.

994. V. Luchangco and K. Koyama, "An Attack on an ID-Based Key Sharing System, *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 262–271.

995. D.J.C. MacKay, "A Free Energy Minimization Framework for Inferring the State of a Shift Register Given the Noisy Output Sequence," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.

996. M.D. MacLaren and G. Marsaglia, "Uniform Random Number Generators," *Journal of the ACM* v. 12, n. 1, Jan 1965, pp. 83–89.

997. D. MacMillan, "Single Chip Encrypts Data at 14Mb/s," *Electronics*, v. 54, n. 12, 16 June 1981, pp. 161–165.

998. R. Madhavan and L.E. Peppard, "A Multiprocessor GaAs RSA Cryptosystem," *Proceedings CCVLSI-89: Canadian Conference on Very Large Scale Integration*, Vancouver, BC, Canada, 22–24 Oct 1989, pp. 115–122.

999. W.E. Madryga, "A High Performance Encryption Algorithm," *Computer Security: A Global Challenge*, Elsevier Science Publishers, 1984, pp. 557–570.

1000. M. Mambo, A. Nishikawa, S. Tsujii, and E. Okamoto, "Efficient Secure Broadcast

Communication System," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 23–33.

1001. M. Mambo, K. Usuda, and E. Okamoto, "Proxy Signatures," *Proceedings of the 1995 Symposium on Cryptography and Information Security (SCIS 95)*, Inuyama, Japan, 24–27 Jan 1995, pp. B1.1.1–17.

1002. W. Mao and C. Boyd, "Towards Formal Analysis of Security Protocols," *Proceedings of the Computer Security Foundations Workshop VI*, IEEE Computer Society Press, 1993, pp. 147–158.

1003. G. Marsaglia and T.A. Bray, "On-Line Random Number Generators and their Use in Combinations," *Communications of the ACM*, v. 11, n. 11, Nov 1968, p. 757–759.

1004. K.M. Martin, "Untrustworthy Participants in Perfect Secret Sharing Schemes," *Cryptography and Coding III*, M.J. Ganley, ed., Oxford: Clarendon Press, 1993, pp. 255–264.

1005. J.L. Massey, "Shift-Register Synthesis and BCH Decoding," *IEEE Transactions on Information Theory*, v. IT-15, n. 1, Jan 1969, pp. 122–127.

1006. J.L. Massey, "Cryptography and System Theory," *Proceedings of the 24th Allerton Conference on Communication, Control, and Computers*, 1–3 Oct 1986, pp. 1–8.

1007. J.L. Massey, "An Introduction to Contemporary Cryptology," *Proceedings of the IEEE*, v. 76, n. 5., May 1988, pp. 533–549.

1008. J.L. Massey, "Contemporary Cryptology: An Introduction," in *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, pp. 1–39.

1009. J.L. Massey, "SAFER K-64: A Byte-Oriented Block-Ciphering Algorithm," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 1–17.

1010. J.L. Massey, "SAFER K-64: One Year Later," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.

1011. J.L. Massey and I. Ingemarsson, "The Rip Van Winkle Cipher—A Simple and Provably Computationally Secure Cipher with a Finite Key," *IEEE International Symposium on Information Theory*, Brighton, UK, May 1985.

1012. J.L. Massey and X. Lai, "Device for Converting a Digital Block and the Use Thereof," International Patent PCT/CH91/00117, 28 Nov 1991.

1013. J.L. Massey and X. Lai, "Device for the Conversion of a Digital Block and Use of Same," U.S. Patent #5,214,703, 25 May 1993.

1014. J.L. Massey and R.A. Rueppel, "Linear Ciphers and Random Sequence Generators with Multiple Clocks," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 74–87.

1015. M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 386–397.

1016. M. Matsui, "Linear Cryptanalysis of DES Cipher (I)," *Proceedings of the 1993 Symposium on Cryptography and Information Security (SCIS 93)*, Shuzenji, Japan, 28–30 Jan 1993, pp. 3C.1–14. (In Japanese.)

1017. M. Matsui, "Linear Cryptanalysis Method for DES Cipher (III)," *Proceedings of the 1994 Symposium on Cryptography and Information Security (SCIS 94)*, Lake Biwa, Japan, 27–29 Jan 1994, pp. 4A.1–11. (In Japanese.)

1018. M. Matsui, "On Correlation Between the Order of the S-Boxes and the Strength of DES," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.

1019. M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994, pp. 1–11.

1020. M. Matsui and A. Yamagishi, "A New Method for Known Plaintext Attack of FEAL Cipher," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 81–91.

1021. T. Matsumoto and H. Imai, "A Class of Asymmetric Crypto-Systems Based on Polynomials Over Finite Rings," *IEEE International Symposium on Information Theory*, 1983, pp. 131–132.

1022. T. Matsumoto and H. Imai, "On the Key Production System: A Practical Solution to the Key Distribution Problem," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 185–193.

1023. T. Matsumoto and H. Imai, "On the Security of Some Key Sharing Schemes (Part

2)," IEICE Japan, Technical Report, ISEC90-28, 1990.

1024. S.M. Matyas, "Digital Signatures—An Overview," *Computer Networks*, v. 3, n. 2, Apr 1979, pp. 87–94.

1025. S.M. Matyas, "Key Handling with Control Vectors," *IBM Systems Journal*, v. 30, n. 2, 1991, pp. 151–174.

1026. S.M. Matyas, A.V. Le, and D.G. Abraham, "A Key Management Scheme Based on Control Vectors," *IBM Systems Journal*, v. 30, n. 2, 1991, pp. 175–191.

1027. S.M. Matyas and C.H. Meyer, "Generation, Distribution, and Installation of Cryptographic Keys," *IBM Systems Journal*, v. 17, n. 2, 1978, pp. 126–137.

1028. S.M. Matyas, C.H. Meyer, and J. Oseas, "Generating Strong One-Way Functions with Cryptographic Algorithm," *IBM Technical Disclosure Bulletin*, v. 27, n. 10A, Mar 1985, pp. 5658–5659.

1029. U.M. Maurer, "Provable Security in Cryptography," Ph.D. dissertation, ETH No. 9260, Swiss Federal Institute of Technology, Zürich, 1990.

1030. U.M. Maurer, "A Provable-Secure Strongly-Randomized Cipher," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1990, pp. 361–373.

1031. U.M. Maurer, "A Universal Statistical Test for Random Bit Generators," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 409–420.

1032. U.M. Maurer, "A Universal Statistical Test for Random Bit Generators," *Journal of Cryptology*, v. 5, n. 2, 1992, pp. 89–106.

1033. U.M. Maurer and J.L. Massey, "Cascade Ciphers: The Importance of Being First," *Journal of Cryptology*, v. 6, n. 1, 1993, pp. 55–61.

1034. U.M. Maurer and J.L. Massey, "Perfect Local Randomness in Pseudo-Random Sequences," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 110–112.

1035. U.M. Maurer and Y. Yacobi, "Non-interactive Public Key Cryptography," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 498–507.

1036. G. Mayhew, "A Low Cost, High Speed Encryption System and Method," *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, 1994, pp. 147–154.

1037. G. Mayhew, R. Frazee, and M. Bianco, "The Kinetic Protection Device," *Proceedings of the 15th National Computer Security Conference*, NIST, 1994, pp. 147–154.

1038. K.S. McCurley, "A Key Distribution System Equivalent to Factoring," *Journal of Cryptology*, v. 1, n. 2, 1988, pp. 95–106.

1039. K.S. McCurley, "The Discrete Logarithm Problem," *Cryptography and Computational Number Theory (Proceedings of the Symposium on Applied Mathematics)*, American Mathematics Society, 1990, pp. 49–74.

1040. K.S. McCurley, open letter from the Sandia National Laboratories on the DSA of the NIST, 7 Nov 1991.

1041. R.J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory," Deep Space Network Progress Report 42–44, Jet Propulsion Laboratory, California Institute of Technology, 1978, pp. 114–116.

1042. R.J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Boston: Kluwer Academic Publishers, 1987.

1043. P. McMahon, "SESAME V2 Public Key and Authorization Extensions to Kerberos," *Proceedings of the Internet Society 1995 Symposium on Network and Distributed Systems Security*, IEEE Computer Society Press, 1995, pp. 114–131.

1044. C.A. Meadows, "A System for the Specification and Analysis of Key Management Protocols," *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 1991, pp. 182–195.

1045. C.A. Meadows, "Applying Formal Methods to the Analysis of a Key Management Protocol," *Journal of Computer Security*, v. 1, n. 1, 1992, pp. 5–35.

1046. C.A. Meadows, "A Model of Computation for the NRL Protocol Analyzer," *Proceedings of the Computer Security Foundations Workshop VII*, IEEE Computer Society Press, 1994, pp. 84–89.

1047. C.A. Meadows, "Formal Verification of Cryptographic Protocols: A Survey," *Advances in Cryptology—ASIACRYPT '94 Proceedings*, Springer-Verlag, 1995, pp. 133–150.

1048. G. Medvinsky and B.C. Neuman, "NetCash: A Design for Practical Electronic Currency on the Internet," *Proceedings of the 1st Annual ACM Conference on Computer and Communications Security,* ACM Press, 1993, pp. 102–106.

1049. G. Medvinsky and B.C. Neuman, "Electronic Currency for the Internet," *Electronic Markets,* v. 3, n. 9/10, Oct 1993, pp. 23–24.

1050. W. Meier, "On the Security of the IDEA Block Cipher," *Advances in Cryptology—EUROCRYPT '93 Proceedings,* Springer-Verlag, 1994, pp. 371–385.

1051. W. Meier and O. Staffelbach, "Fast Correlation Attacks on Stream Ciphers," *Journal of Cryptology,* v. 1, n. 3, 1989, pp. 159–176.

1052. W. Meier and O. Staffelbach, "Analysis of Pseudo Random Sequences Generated by Cellular Automata," *Advances in Cryptology—EUROCRYPT '91 Proceedings,* Springer-Verlag, 1991, pp. 186–199.

1053. W. Meier and O. Staffelbach, "Correlation Properties of Combiners with Memory in Stream Ciphers," *Advances in Cryptology—EUROCRYPT '90 Proceedings,* Springer-Verlag, 1991, pp. 204–213.

1054. W. Meier and O. Staffelbach, "Correlation Properties of Combiners with Memory in Stream Ciphers," *Journal of Cryptology,* v. 5, n. 1, 1992, pp. 67–86.

1055. W. Meier and O. Staffelbach, "The Self-Shrinking Generator," *Communications and Cryptography: Two Sides of One Tapestry,* R.E. Blahut et al., eds., Kluwer Adademic Publishers, 1994, pp. 287–295.

1056. J. Meijers, "Algebraic-Coded Cryptosystems," Master's thesis, Technical University Eindhoven, 1990.

1057. J. Meijers and J. van Tilburg, "On the Rao-Nam Private-Key Cryptosystem Using Linear Codes," *International Symposium on Information Theory,* Budapest, Hungary, 1991.

1058. J. Meijers and J. van Tilburg, "An Improved ST-Attack on the Rao-Nam Private-Key Cryptosystem," *International Conference on Finite Fields, Coding Theory, and Advances in Communications and Computing,* Las Vegas, NV, 1991.

1059. A. Menezes, *Elliptic Curve Public Key Cryptosystems,* Kluwer Academic Publishers, 1993.

1060. A. Menezes, ed., *Applications of Finite Fields,* Kluwer Academic Publishers, 1993.

1061. A. Menezes and S.A. Vanstone, "Elliptic Curve Cryptosystems and Their Implementations," *Journal of Cryptology,* v. 6, n. 4, 1993, pp. 209–224.

1062. A. Menezes and S.A. Vanstone, "The Implementation of Elliptic Curve Cryptosystems," *Advances in Cryptology—AUSCRYPT '90 Proceedings,* Springer-Verlag, 1990, pp. 2–13.

1063. R. Menicocci, "Short Gollmann Cascade Generators May Be Insecure," *Codes and Ciphers,* Institute of Mathematics and its Applications, 1995, pp. 281–297.

1064. R.C. Merkle, "Secure Communication Over Insecure Channels," *Communications of the ACM,* v. 21, n. 4, 1978, pp. 294–299.

1065. R.C. Merkle, "Secrecy, Authentication, and Public Key Systems," Ph.D. dissertation, Stanford University, 1979.

1066. R.C. Merkle, "Method of Providing Digital Signatures," U.S. Patent #4,309,569, 5 Jan 1982.

1067. R.C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," *Advances in Cryptology—CRYPTO '87 Proceedings,* Springer-Verlag, 1988, pp. 369–378.

1068. R.C. Merkle, "A Certified Digital Signature," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 218–238.

1069. R.C. Merkle, "One Way Hash Functions and DES," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 428–446.

1070. R.C. Merkle, "A Fast Software One-Way Hash Function," *Journal of Cryptology,* v. 3, n. 1, 1990, pp. 43–58.

1071. R.C. Merkle, "Fast Software Encryption Functions," *Advances in Cryptology—CRYPTO '90 Proceedings,* Springer-Verlag, 1991, pp. 476–501.

1072. R.C. Merkle, "Method and Apparatus for Data Encryption," U.S. Patent #5,003,597, 26 Mar 1991.

1073. R.C. Merkle, personal communication, 1993.

1074. R.C. Merkle and M. Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks," *IEEE Transactions on Infor-*

mation Theory, v. 24, n. 5, Sep 1978, pp. 525–530.

1075. R.C. Merkle and M. Hellman, "On the Security of Multiple Encryption," *Communications of the ACM,* v. 24, n. 7, 1981, pp. 465–467.

1076. M. Merritt, "Cryptographic Protocols," Ph.D. dissertation, Georgia Institute of Technology, GIT-ICS-83/6, Feb 1983.

1077. M. Merritt, "Towards a Theory of Cryptographic Systems: A Critique of Crypto-Complexity," *Distributed Computing and Cryptography,* J. Feigenbaum and M. Merritt, eds., American Mathematical Society, 1991, pp. 203–212.

1078. C.H. Meyer, "Ciphertext/Plaintext and Ciphertext/Key Dependencies vs. Number of Rounds for Data Encryption Standard," *AFIPS Conference Proceedings, 47,* 1978, pp. 1119–1126.

1079. C.H. Meyer, "Cryptography—A State of the Art Review," *Proceedings of Compeuro '89, VLSI and Computer Peripherals, 3rd Annual European Computer Conference,* IEEE Press, 1989, pp. 150–154.

1080. C.H. Meyer and S.M. Matyas, *Cryptography: A New Dimension in Computer Data Security,* New York: John Wiley & Sons, 1982.

1081. C.H. Meyer and M. Schilling, "Secure Program Load with Manipulation Detection Code," *Proceedings of Securicom '88,* 1988, pp. 111–130.

1082. C.H. Meyer and W.L. Tuchman, "Pseudo-Random Codes Can Be Cracked," *Electronic Design,* v. 23, Nov 1972.

1083. C.H. Meyer and W.L. Tuchman, "Design Considerations for Cryptography," *Proceedings of the NCC,* v. 42, Montvale, NJ: AFIPS Press, Nov 1979, pp. 594–597.

1084. S. Micali, "Fair Public-Key Cryptosystems," *Advances in Cryptology—CRYPTO '92 Proceedings,* Springer-Verlag, 1993, pp. 113–138.

1085. S. Micali, "Fair Cryptosystems," MIT/LCS/TR-579.b, MIT Laboratory for Computer Science, Nov 1993.

1086. S. Micali, "Fair Cryptosystems and Methods for Use," U.S. Patent #5,276,737, 4 Jan 1994.

1087. S. Micali, "Fair Cryptosystems and Methods for Use," U.S. Patent #5,315,658, 24 May 1994.

1088. S. Micali and A. Shamir, "An Improvement on the Fiat-Shamir Identification and Signature Scheme," *Advances in Cryptology—CRYPTO '88 Proceedings,* Springer-Verlag, 1990, pp. 244–247.

1089. M.J. Mihajlevič, "A Correlation Attack on the Binary Sequence Generators with Time-Varying Output Function," *Advances in Cryptology—ASIACRYPT '94 Proceedings,* Springer-Verlag, 1995, pp. 67–79.

1090. M.J. Mihajlevič and J.D. Golič, "A Fast Iterative Algorithm for a Shift Register Internal State Reconstruction Given the Noisy Output Sequence," *Advances in Cryptology—AUSCRYPT '90 Proceedings,* Springer-Verlag, 1990, pp. 165–175.

1091. M.J. Mihajlevič and J.D. Golič, "Convergence of a Bayesian Iterative Error-Correction Procedure to a Noisy Shift Register Sequence," *Advances in Cryptology—EUROCRYPT '92 Proceedings,* Springer-Verlag, 1993, pp. 124–137.

1092. J.K. Millen, S.C. Clark, and S.B. Freedman, "The Interrogator: Protocol Security Analysis," *IEEE Transactions on Software Engineering,* v. SE-13, n. 2, Feb 1987, pp. 274–288.

1093. G.L. Miller, "Riemann's Hypothesis and Tests for Primality," *Journal of Computer Systems Science,* v. 13, n. 3, Dec 1976, pp. 300–317.

1094. S.P. Miller, B.C. Neuman, J.I. Schiller, and J.H. Saltzer, "Section E.2.1: Kerberos Authentication and Authorization System," MIT Project Athena, Dec 1987.

1095. V.S. Miller, "Use of Elliptic Curves in Cryptography," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 417–426.

1096. M. Minsky, *Computation: Finite and Infinite Machines,* Englewood Cliffs, NJ: Prentice-Hall, 1967.

1097. C.J. Mitchell, "Authenticating Multi-Cast Internet Electronic Mail Messages Using a Bidirectional MAC Is Insecure," draft manuscript, 1990.

1098. C.J. Mitchell, "Enumerating Boolean Functions of Cryptographic Significance," *Journal of Cryptology,* v. 2, n. 3, 1990, pp. 155–170.

1099. C.J. Mitchell, F. Piper, and P. Wild, "Digital Signatures," *Contemporary Cryptology: The Science of Information Integrity,* G.J. Simmons, ed., IEEE Press, 1991, pp. 325–378.

1100. C.J. Mitchell, M. Walker, and D. Rush, "CCITT/ISO Standards for Secure Message

Handling," *IEEE Journal on Selected Areas in Communications*, v. 7, n. 4, May 1989, pp. 517–524.

1101. S. Miyaguchi, "Fast Encryption Algorithm for the RSA Cryptographic System," *Proceedings of Compcon 82*, IEEE Press, pp. 672–678.

1102. S. Miyaguchi, "The FEAL-8 Cryptosystem and Call for Attack," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 624–627.

1103. S. Miyaguchi, "Expansion of the FEAL Cipher," *NTT Review*, v. 2, n. 6, Nov 1990.

1104. S. Miyaguchi, "The FEAL Cipher Family," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 627–638.

1105. S. Miyaguchi, K. Ohta, and M. Iwata, "128-bit Hash Function (*N*-Hash)," *Proceedings of SECURICOM '90*, 1990, pp. 127–137.

1106. S. Miyaguchi, K. Ohta, and M. Iwata, "128-bit Hash Function (*N*-Hash)," *NTT Review*, v. 2, n. 6, Nov 1990, pp. 128–132.

1107. S. Miyaguchi, K. Ohta, and M. Iwata, "Confirmation that Some Hash Functions Are Not Collision Free," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 326–343.

1108. S. Miyaguchi, A. Shiraishi, and A. Shimizu, "Fast Data Encipherment Algorithm FEAL-8," *Review of the Electrical Communication Laboratories*, v. 36, n. 4, 1988.

1109. H. Miyano, "Differential Cryptanalysis on CALC and Its Evaluation," *Proceedings of the 1992 Symposium on Cryptography and Information Security (SCIS 92)*, Tateshina, Japan, 2–4 Apr 1992, pp. 7B.1–8.

1110. R. Molva, G. Tsudik, E. van Herreweghen, and S. Zatti, "KryptoKnight Authentication and Key Distribution System," *Proceedings of European Symposium on Research in Computer Security*, Toulouse, France, Nov 1992.

1111. P.L. Montgomery, "Modular Multiplication without Trial Division," *Mathematics of Computation*, v. 44, n. 170, 1985, pp. 519–521.

1112. P.L. Montgomery, "Speeding the Pollard and Elliptic Curve Methods of Factorization," *Mathematics of Computation*, v. 48, n. 177, Jan 1987, pp. 243–264.

1113. P.L. Montgomery and R. Silverman, "An FFT Extension to the p-1 Factoring Algorithm," *Mathematics of Computation*, v. 54, n. 190, 1990, pp. 839–854.

1114. J.H. Moore, "Protocol Failures in Cryptosystems," *Proceedings of the IEEE*, v. 76, n. 5, May 1988.

1115. J.H. Moore, "Protocol Failures in Cryptosystems," in *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, pp. 541–558.

1116. J.H. Moore and G.J. Simmons, "Cycle Structure of the DES with Weak and Semi-Weak Keys," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 3–32.

1117. T. Moriyasu, M. Morii, and M. Kasahara, "Nonlinear Pseudorandom Number Generator with Dynamic Structure and Its Properties," *Proceedings of the 1994 Symposium on Cryptography and Information Security (SCIS 94)*, Biwako, Japan, 27–29 Jan 1994, pp. 8A.1–11.

1118. R. Morris, "The Data Encryption Standard—Retrospective and Prospects," *IEEE Communications Magazine*, v. 16, n. 6, Nov 1978, pp. 11–14.

1119. R. Morris, remarks at the 1993 Cambridge Protocols Workshop, 1993.

1120. R. Morris, N.J.A. Sloane, and A.D. Wyner, "Assessment of the NBS Proposed Data Encryption Standard," *Cryptologia*, v. 1, n. 3, Jul 1977, pp. 281–291.

1121. R. Morris and K. Thompson, "Password Security: A Case History," *Communications of the ACM*, v. 22, n. 11, Nov 1979, pp. 594–597.

1122. S.B. Morris, "Escrow Encryption," lecture at MIT Laboratory for Computer Science, 2 Jun 1994.

1123. M.N. Morrison and J. Brillhart, "A Method of Factoring and the Factorization of F_7," *Mathematics of Computation*, v. 29, n. 129, Jan 1975, pp. 183–205.

1124. L.E. Moser, "A Logic of Knowledge and Belief for Reasoning About Computer Security," *Proceedings of the Computer Security Foundations Workshop II*, IEEE Computer Society Press, 1989, pp. 57–63.

1125. Motorola Government Electronics Division, *Advanced Techniques in Network Security*, Scottsdale, AZ, 1977.

1126. W.B. Müller, "Polynomial Functions in Modern Cryptology," *Contributions to General Algebra 3: Proceedings of the*

Vienna Conference, Vienna: Verlag Hölder-Pichler-Tempsky, 1985, pp. 7–32.

1127. W.B. Müller and W. Nöbauer, "Some Remarks on Public-Key Cryptography," *Studia Scientiarum Mathematicarum Hungarica,* v. 16, 1981, pp. 71–76.

1128. W.B. Müller and W. Nöbauer, "Cryptanalysis of the Dickson Scheme," *Advances in Cryptology—EUROCRYPT '85 Proceedings,* Springer-Verlag, 1986, pp. 50–61.

1129. C. Muller-Scholer, "A Microprocessor-Based Cryptoprocessor," *IEEE Micro,* Oct 1983, pp. 5–15.

1130. R.C. Mullin, E. Nemeth, and N. Weidenhofer, "Will Public Key Cryptosystems Live Up to Their Expectations?—HEP Implementation of the Discrete Log Codebreaker," *ICPP 85,* pp. 193–196.

1131. Y. Murakami and S. Kasahara, "An ID-Based Key Distribution Scheme," IEICE Japan, Technical Report, ISEC90-26, 1990.

1132. S. Murphy, "The Cryptanalysis of FEAL-4 with 20 Chosen Plaintexts," *Journal of Cryptology,* v. 2, n. 3, 1990, pp. 145–154.

1133. E.D. Myers, "STU-III—Multilevel Secure Computer Interface," *Proceedings of the Tenth Annual Computer Security Applications Conference,* IEEE Computer Society Press, 1994, pp. 170–179.

1134. D. Naccache, "Can O.S.S. be Repaired? Proposal for a New Practical Signature Scheme," *Advances in Cryptology—EUROCRYPT '93 Proceedings,* Springer-Verlag, 1994, pp. 233–239.

1135. D. Naccache, D. M'Raïhi, D. Raphaeli, and S. Vaudenay, "Can D.S.A. be Improved? Complexity Trade-Offs with the Digital Signature Standard," *Advances in Cryptology—EUROCRYPT '94 Proceedings,* Springer-Verlag, 1995, to appear.

1136. Y. Nakao, T. Kaneko, K. Koyama, and R. Terada, "A Study on the Security of RDES-1 Cryptosystem against Linear Cryptanalysis," *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography,* Inuyama, Japan, 24–27 Jan 1995, pp. 163–172.

1137. M. Naor, "Bit Commitment Using Pseudo-Randomness," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 128–136.

1138. M. Naor and M. Yung, "Universal One-Way Hash Functions and Their Cryptographic Application," *Proceedings of the 21st Annual ACM Symposium on the Theory of Computing,* 1989, pp. 33–43.

1139. National Bureau of Standards, "Report of the Workshop on Estimation of Significant Advances in Computer Technology," NBSIR76-1189, National Bureau of Standards, U.S. Department of Commerce, 21–22 Sep 1976, Dec 1977.

1140. National Bureau of Standards, NBS FIPS PUB 46, "Data Encryption Standard," National Bureau of Standards, U.S. Department of Commerce, Jan 1977.

1141. National Bureau of Standards, NBS FIPS PUB 46-1, "Data Encryption Standard," U.S. Department of Commerce, Jan 1988.

1142. National Bureau of Standards, NBS FIPS PUB 74, "Guidelines for Implementing and Using the NBS Data Encryption Standard," U.S. Department of Commerce, Apr 1981.

1143. National Bureau of Standards, NBS FIPS PUB 81, "DES Modes of Operation," U.S. Department of Commerce, Dec 1980.

1144. National Bureau of Standards, NBS FIPS PUB 112, "Password Usage," U.S. Department of Commerce, May 1985.

1145. National Bureau of Standards, NBS FIPS PUB 113, "Computer Data Authentication," U.S. Department of Commerce, May 1985.

1146. National Computer Security Center, "Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria," NCSC-TG-005 Version 1, Jul 1987.

1147. National Computer Security Center, "Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria," NCSC-TG-021 Version 1, Apr 1991.

1148. National Computer Security Center, "A Guide to Understanding Data Remembrance in Automated Information Systems," NCSC-TG-025 Version 2, Sep 1991.

1149. National Institute of Standards and Technology, NIST FIPS PUB XX, "Digital Signature Standard," U.S. Department of Commerce, DRAFT, 19 Aug 1991.

1150. National Institute of Standards and Technology, NIST FIPS PUB 46-2, "Data Encryption Standard," U.S. Department of Commerce, Dec 93.

1151. National Institute of Standards and Technology, NIST FIPS PUB 171, "Key Management Using X9.17," U.S. Department of Commerce, Apr 92.

1152. National Institute of Standards and Technology, NIST FIPS PUB 180, "Secure Hash Standard," U.S. Department of Commerce, May 93.

1153. National Institute of Standards and Technology, NIST FIPS PUB 185, "Escrowed Encryption Standard," U.S. Department of Commerce, Feb 94.

1154. National Institute of Standards and Technology, NIST FIPS PUB 186, "Digital Signature Standard," U.S. Department of Commerce, May 1994.

1155. National Institute of Standards and Technology, "Clipper Chip Technology," 30 Apr 1993.

1156. National Institute of Standards and Technology, "Capstone Chip Technology," 30 Apr 1993.

1157. J. Nechvatal, "Public Key Cryptography," NIST Special Publication 800-2, National Institute of Standards and Technology, U.S. Department of Commerce, Apr 1991.

1158. J. Nechvatal, "Public Key Cryptography," *Contemporary Cryptology: The Science of Information Integrity,* G.J. Simmons, ed., IEEE Press, 1992, pp. 177–288.

1159. R.M. Needham and M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM,* v. 21, n. 12, Dec 1978, pp. 993–999.

1160. R.M. Needham and M.D. Schroeder, "Authentication Revisited," *Operating Systems Review,* v. 21, n. 1, 1987, p. 7.

1161. D.M. Nessett, "A Critique of the Burrows, Abadi, and Needham Logic," *Operating System Review,* v. 20, n. 2, Apr 1990, pp. 35–38.

1162. B.C. Neuman and S. Stubblebine, "A Note on the Use of Timestamps as Nonces," *Operating Systems Review,* v. 27, n. 2, Apr 1993, pp. 10–14.

1163. B.C. Neuman and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," *IEEE Communications Magazine,* v. 32, n. 9, Sep 1994, pp. 33–38.

1164. L. Neuwirth, "Statement of Lee Neuwirth of Cylink on HR145," submitted to congressional committees considering HR145, Feb 1987.

1165. D.B. Newman, Jr. and R.L. Pickholtz, "Cryptography in the Private Sector," *IEEE Communications Magazine,* v. 24, n. 8, Aug 1986, pp. 7–10.

1166. H. Niederreiter, "A Public-Key Cryptosystem Based on Shift Register Sequences," *Advances in Cryptology—EUROCRYPT '85 Proceedings,* Springer-Verlag, 1986, pp. 35–39.

1167. H. Niederreiter, "Knapsack-Type Cryptosystems and Algebraic Coding Theory," *Problems of Control and Information Theory,* v. 15, n. 2, 1986, pp. 159–166.

1168. H. Niederreiter, "The Linear Complexity Profile and the Jump Complexity of Keystream Sequences," *Advances in Cryptology—EUROCRYPT '90 Proceedings,* Springer-Verlag, 1991, pp. 174–188.

1169. V. Niemi, "A New Trapdoor in Knapsacks," *Advances in Cryptology—EUROCRYPT '90 Proceedings,* Springer-Verlag, 1991, pp. 405–411.

1170. V. Niemi and A. Renvall, "How to Prevent Buying of Voters in Computer Elections," *Advances in Cryptology—ASIACRYPT '94 Proceedings,* Springer-Verlag, 1995, pp. 164–170.

1171. I. Niven and H.A. Zuckerman, *An Introduction to the Theory of Numbers,* New York: John Wiley & Sons, 1972.

1172. R. Nöbauer, "Cryptanalysis of the Rédei Scheme," *Contributions to General Algebra 3: Proceedings of the Vienna Conference,* Verlag Hölder-Pichler-Tempsky, Vienna, 1985, pp. 255–264.

1173. R. Nöbauer, "Cryptanalysis of a Public-Key Cryptosystem Based on Dickson-Polynomials," *Mathematica Slovaca,* v. 38, n. 4, 1988, pp. 309–323.

1174. K. Noguchi, H. Ashiya, Y. Sano, and T. Kaneko, "A Study on Differential Attack of MBAL Cryptosystem," *Proceedings of the 1994 Symposium on Cryptography and Information Security (SCIS 94),* Lake Biwa, Japan, 27–29 Jan 1994, pp. 14B.1–7. (In Japanese.)

1175. H. Nurmi, A. Salomaa, and L. Santean, "Secret Ballot Elections in Computer Networks," *Computers & Security,* v. 10, 1991, pp. 553–560.

1176. K. Nyberg, "Construction of Bent Functions and Difference Sets," *Advances in Cryptology—EUROCRYPT '91 Proceedings,* Springer-Verlag, 1991, pp. 151–160.

1177. K. Nyberg, "Perfect Nonlinear S-Boxes," *Advances in Cryptology—EUROCRYPT '91 Proceedings,* Springer-Verlag, 1991, pp. 378–386.

1178. K. Nyberg, "On the Construction of Highly Nonlinear Permutations," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1991, pp. 92–98.

1179. K. Nyberg, "Differentially Uniform Mappings for Cryptography," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 55–64.

1180. K. Nyberg, "Provable Security against Differential Cryptanalysis," presented at the rump session of Eurocrypt '94, May 1994.

1181. K. Nyberg and L.R. Knudsen, "Provable Security against Differential Cryptanalysis," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, pp. 566–574.

1182. K. Nyberg and L.R. Knudsen, "Provable Security against Differential Cryptanalysis," *Journal of Cryptology*, v. 8, n. 1, 1995, pp. 27–37.

1183. K. Nyberg and R.A. Rueppel, "A New Signature Scheme Based on the DSA Giving Message Recovery," *1st ACM Conference on Computer and Communications Security*, ACM Press, 1993, pp. 58–61.

1184. K. Nyberg and R.A. Rueppel, "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.

1185. L. O'Connor, "Enumerating Nondegenerate Permutations," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 368–377.

1186. L. O'Connor, "On the Distribution of Characteristics in Bijective Mappings," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 360–370.

1187. L. O'Connor, "On the Distribution of Characteristics in Composite Permutations," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 403–412.

1188. L. O'Connor and A. Klapper, "Algebraic Nonlinearity and Its Application to Cryptography," *Journal of Cryptology*, v. 7, n. 3, 1994, pp. 133–151.

1189. A. Odlyzko, "Discrete Logarithms in Finite Fields and Their Cryptographic Significance," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 224–314.

1190. A. Odlyzko, "Progress in Integer Factorization and Discrete Logarithms," unpublished manuscript, Feb 1995.

1191. Office of Technology Assessment, U.S. Congress, "Defending Secrets, Sharing Data: New Locks and Keys for Electronic Communication," OTA-CIT-310, Washington, D.C.: U.S. Government Printing Office, Oct 1987.

1192. B. O'Higgins, W. Diffie, L. Strawczynski, and R. de Hoog, "Encryption and ISDN—a Natural Fit," *Proceedings of the 1987 International Switching Symposium*, 1987, pp. 863–869.

1193. Y. Ohnishi, "A Study on Data Security," Master's thesis, Tohuku University, Japan, 1988. (In Japanese.)

1194. K. Ohta, "A Secure and Efficient Encrypted Broadcast Communication System Using a Public Master Key," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. J70-D, n. 8, Aug 1987, pp. 1616–1624.

1195. K. Ohta, "An Electrical Voting Scheme Using a Single Administrator," *IEICE Spring National Convention*, A-294, 1988, v. 1, p. 296. (In Japanese.)

1196. K. Ohta, "Identity-based Authentication Schemes Using the RSA Cryptosystem," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. J72D-II, n. 8, Aug 1989, pp. 612–620.

1197. K. Ohta and M. Matsui, "Differential Attack on Message Authentication Codes," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 200–223.

1198. K. Ohta and T. Okamoto, "Practical Extension of Fiat-Shamir Scheme," *Electronics Letters*, v. 24, n. 15, 1988, pp. 955–956.

1199. K. Ohta and T. Okamoto, "A Modification of the Fiat-Shamir Scheme," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 232–243.

1200. K. Ohta and T. Okamoto, "A Digital Multisignature Scheme Based on the Fiat-Shamir Scheme," *Advances in Cryptology—ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 139–148.

1201. K. Ohta, T. Okamoto and K. Koyama, "Membership Authentication for Hierarchy Multigroups Using the Extended Fiat-Shamir Scheme," *Advances in Cryptol-*

ogy—EUROCRYPT '90 Proceedings, Springer-Verlag, 1991, pp. 446–457.

1202. E. Okamoto and K. Tanaka, "Key Distribution Based on Identification Information," *IEEE Journal on Selected Areas in Communication*, v. 7, n. 4, May 1989, pp. 481–485.
1203. T. Okamoto, "Fast Public-Key Cryptosystems Using Congruent Polynomial Equations," *Electronics Letters*, v. 22, n. 11, 1986, pp. 581–582.
1204. T. Okamoto, "Modification of a Public-Key Cryptosystem," *Electronics Letters*, v. 23, n. 16, 1987, pp. 814–815.
1205. T. Okamoto, "A Fast Signature Scheme Based on Congruential Polynomial Operations," *IEEE Transactions on Information Theory*, v. 36, n. 1, 1990, pp. 47–53.
1206. T. Okamoto, "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, pp. 31–53.
1207. T. Okamoto, A. Fujioka, and E. Fujisaki, "An Efficient Digital Signature Scheme Based on Elliptic Curve over the Ring Z_n," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, 1993, pp. 54–65.
1208. T. Okamoto, S. Miyaguchi, A. Shiraishi, and T. Kawoaka, "Signed Document Transmission System," U.S. Patent #4,625,076, 25 Nov 1986.
1209. T. Okamoto and K. Ohta, "Disposable Zero-Knowledge Authentication and Their Applications to Untraceable Electronic Cash," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 134–149.
1210. T. Okamoto and K. Ohta, "How to Utilize the Randomness of Zero-Knowledge Proofs," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 456–475.
1211. T. Okamoto and K. Ohta, "Universal Electronic Cash," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 324–337.
1212. T. Okamoto and K. Ohta, "Survey of Digital Signature Schemes," *Proceedings of the Third Symposium on State and Progress of Research in Cryptography*, Fondazone Ugo Bordoni, Rome, 1993, pp. 17–29.
1213. T. Okamoto and K. Ohta, "Designated Confirmer Signatures Using Trapdoor Functions," *Proceedings of the 1994 Symposium on Cryptography and Information Security (SCIS 94)*, Lake Biwa, Japan, 27–29 Jan 1994, pp. 16B.1–11.
1214. T. Okamoto and K. Sakurai, "Efficient Algorithms for the Construction of Hyperelliptic Cryptosystems," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 267–278.
1215. T. Okamoto and A. Shiraishi, "A Fast Signature Scheme Based on Quadratic Inequalities," *Proceedings of the 1985 Symposium on Security and Privacy*, IEEE, Apr 1985, pp. 123–132.
1216. J.D. Olsen, R.A. Scholtz, and L.R. Welch, "Bent Function Sequences," *IEEE Transactions on Information Theory*, v. IT-28, n. 6, Nov 1982, pp. 858–864.
1217. H. Ong and C.P. Schnorr, "Signatures through Approximate Representations by Quadratic Forms," *Advances in Cryptology: Proceedings of Crypto 83*, Plenum Press, 1984.
1218. H. Ong and C.P. Schnorr, "Fast Signature Generation with a Fiat Shamir-Like Scheme," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 432–440.
1219. H. Ong, C.P. Schnorr, and A. Shamir, "An Efficient Signature Scheme Based on Polynomial Equations," *Proceedings of the 16th Annual Symposium on the Theory of Computing*, 1984, pp. 208–216.
1220. H. Ong, C.P. Schnorr, and A. Shamir, "Efficient Signature Schemes Based on Polynomial Equations," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 37–46.
1221. Open Shop Information Services, *OSIS Security Aspects*, OSIS European Working Group, WG1, final report, Oct 1985.
1222. G.A. Orton, M.P. Roy, P.A. Scott, L.E. Peppard, and S.E. Tavares, "VLSI Implementation of Public-Key Encryption Algorithms," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 277–301.
1223. H. Orup, E. Svendsen, and E. Andreasen, "VICTOR—An Efficient RSA Hardware Implementation," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 245–252.
1224. D. Otway and O. Rees, "Efficient and Timely Mutual Authentication," *Operating Systems Review*, v. 21, n. 1, 1987, pp. 8–10.

1225. G. Pagels-Fick, "Implementation Issues for Master Key Distribution and Protected Keyload Procedures," *Computers and Security: A Global Challenge, Proceedings of IFIP/SEC '83,* North Holland: Elsevier Science Publishers, 1984, pp. 381–390.

1226. C.M. Papadimitriou, *Computational Complexity,* Addison-Wesley, 1994.

1227. C.S. Park, "Improving Code Rate of McEliece's Public-key Cryptosystem," *Electronics Letters,* v. 25, n. 21, 12 Oct 1989, pp. 1466–1467.

1228. S. Park, Y. Kim, S. Lee, and K. Kim, "Attacks on Tanaka's Non-interactive Key Sharing Scheme," *Proceedings of the 1995 Symposium on Cryptography and Information Security (SCIS 95),* Inuyama, Japan, 24–27 Jan 1995, pp. B3.4.1–4.

1229. S.J. Park, K.H. Lee, and D.H. Won, "An Entrusted Undeniable Signature," *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography,* Inuyama, Japan, 24–27 Jan 1995, pp. 120–126.

1230. S.J. Park, K.H. Lee, and D.H. Won, "A Practical Group Signature," *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography,* Inuyama, Japan, 24–27 Jan 1995, pp. 127–133.

1231. S.K. Park and K.W. Miller, "Random Number Generators: Good Ones Are Hard to Find," *Communications of the ACM,* v. 31, n. 10, Oct 1988, pp. 1192–1201.

1232. J. Patarin, "How to Find and Avoid Collisions for the Knapsack Hash Function," *Advances in Cryptology—EUROCRYPT '93 Proceedings,* Springer-Verlag, 1994, pp. 305–317.

1233. W. Patterson, *Mathematical Cryptology for Computer Scientists and Mathematicians,* Totowa, N.J.: Rowman & Littlefield, 1987.

1234. W.H. Payne, "Public Key Cryptography Is Easy to Break," William H. Payne, unpublished manuscript, 16 Oct 90.

1235. T.P. Pederson, "Distributed Provers with Applications to Undeniable Signatures," *Advances in Cryptology—EUROCRYPT '91 Proceedings,* Springer-Verlag, 1991, pp. 221–242.

1236. S. Peleg and A. Rosenfield, "Breaking Substitution Ciphers Using a Relaxation Algorithm," *Communications of the ACM,* v. 22, n. 11, Nov 1979, pp. 598–605.

1237. R. Peralta, "Simultaneous Security of Bits in the Discrete Log," *Advances in Cryptology—EUROCRYPT '85,* Springer-Verlag, 1986, pp. 62–72.

1238. I. Peterson, "Monte Carlo Physics: A Cautionary Lesson," *Science News,* v. 142, n. 25, 19 Dec 1992, p. 422.

1239. B. Pfitzmann, "Fail-Stop Signatures: Principles and Applications," *Proceedings of COMPUSEC '91, Eighth World Conference on Computer Security, Audit, and Control,* Elsevier Science Publishers, 1991, pp. 125–134.

1240. B. Pfitzmann and M. Waidner, "Formal Aspects of Fail-Stop Signatures," Fakultät für Informatik, University Karlsruhe, Report 22/90, 1990.

1241. B. Pfitzmann and M. Waidner, "Fail-Stop Signatures and Their Application," *Securicom '91,* 1991, pp. 145–160.

1242. B. Pfitzmann and M. Waidner, "Unconditional Concealment with Cryptographic Ruggedness," *VIS '91 Verlassliche Informationsysteme Proceedings,* Darmstadt, Germany, 13–15 March 1991, pp. 3-2-320. (In German.)

1243. B. Pfitzmann and M. Waidner, "How to Break and Repair a 'Provably Secure' Untraceable Payment System," *Advances in Cryptology—CRYPTO '91 Proceedings,* Springer-Verlag, 1992, pp. 338–350.

1244. C.P. Pfleeger, *Security in Computing,* Englewood Cliffs, N.J.: Prentice-Hall, 1989.

1245. S.J.D. Phoenix and P.D. Townsend, "Quantum Cryptography and Secure Optical Communication," *BT Technology Journal,* v. 11, n. 2, Apr 1993, pp. 65–75.

1246. J. Pieprzyk, "On Public-Key Cryptosystems Built Using Polynomial Rings," *Advances in Cryptology—EUROCRYPT '85,* Springer-Verlag, 1986, pp. 73–80.

1247. J. Pieprzyk, "Error Propagation Property and Applications in Cryptography," *IEE Proceedings-E, Computers and Digital Techniques,* v. 136, n. 4, Jul 1989, pp. 262–270.

1248. D. Pinkas, T. Parker, and P. Kaijser, "SESAME: An Introduction," Issue 1.2, Bull, ICL, and SNI, Sep 1993.

1249. F. Piper, "Stream Ciphers," *Elektrotechnic und Maschinenbau,* v. 104, n. 12, 1987, pp. 564–568.

1250. V.S. Pless, "Encryption Schemes for Computer Confidentiality," *IEEE Transactions on Computing,* v. C-26, n. 11, Nov 1977, pp. 1133–1136.

1251. J.B. Plumstead, "Inferring a Sequence Generated by a Linear Congruence," *Proceedings of the 23rd IEEE Symposium on the Foundations of Computer Science*, 1982, pp. 153–159.

1252. R. Poet, "The Design of Special Purpose Hardware to Factor Large Integers," *Computer Physics Communications*, v. 37, 1985, pp. 337–341.

1253. S.C. Pohlig and M.E. Hellman, "An Improved Algorithm for Computing Logarithms in GF(p) and Its Cryptographic Significance," *IEEE Transactions on Information Theory*, v. 24, n. 1, Jan 1978, pp. 106–111.

1254. J.M. Pollard, "A Monte Carlo Method for Factorization," *BIT*, v. 15, 1975, pp. 331–334.

1255. J.M. Pollard and C.P. Schnorr, "An Efficient Solution of the Congruence $x^2 + ky^2 = m \pmod{n}$," *IEEE Transactions on Information Theory*, v. IT-33, n. 5, Sep 1987, pp. 702–709.

1256. C. Pomerance, "Recent Developments in Primality Testing," *The Mathematical Intelligencer*, v. 3, n. 3, 1981, pp. 97–105.

1257. C. Pomerance, "The Quadratic Sieve Factoring Algorithm," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, 169–182.

1258. C. Pomerance, "Fast, Rigorous Factorization and Discrete Logarithm Algorithms," *Discrete Algorithms and Complexity*, New York: Academic Press, 1987, pp. 119–143.

1259. C. Pomerance, J.W. Smith, and R. Tuler, "A Pipe-Line Architecture for Factoring Large Integers with the Quadratic Sieve Algorithm," *SIAM Journal on Computing*, v. 17, n. 2, Apr 1988, pp. 387–403.

1260. G.J. Popek and C.S. Kline, "Encryption and Secure Computer Networks," *ACM Computing Surveys*, v. 11, n. 4, Dec 1979, pp. 331–356.

1261. F. Pratt, *Secret and Urgent*, Blue Ribbon Books, 1942.

1262. B. Preneel, "Analysis and Design of Cryptographic Hash Functions," Ph.D. dissertation, Katholieke Universiteit Leuven, Jan 1993.

1263. B. Preneel, "Differential Cryptanalysis of Hash Functions Based on Block Ciphers," *Proceedings of the 1st ACM Conference on Computer and Communications Security*, 1993, pp. 183–188.

1264. B. Preneel, "Cryptographic Hash Functions," *European Transactions on Telecommunications*, v 5, n. 4, Jul/Aug 1994, pp. 431–448.

1265. B. Preneel, personal communication, 1995.

1266. B. Preneel, A. Bosselaers, R. Govaerts, and J. Vandewalle, "Collision-Free Hash Functions Based on Block Cipher Algorithms," *Proceedings of the 1989 Carnahan Conference on Security Technology*, 1989, pp. 203–210.

1267. B. Preneel, R. Govaerts, and J. Vandewalle, "An Attack on Two Hash Functions by Zheng-Matsumoto-Imai," *Advances in Cryptology—ASIACRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 535–538.

1268. B. Preneel, R. Govaerts, and J. Vandewalle, "Hash Functions Based on Block Ciphers: A Synthetic Approach," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 368–378.

1269. B. Preneel, M. Nuttin, V. Rijmen, and J. Buelens, "Cryptanalysis of the CFB mode of the DES with a Reduced Number of Rounds," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 212–223.

1270. B. Preneel and V. Rijmen, "On Using Maximum Likelihood to Optimize Recent Cryptanalytic Techniques," presented at the rump session of EUROCRYPT '94, May 1994.

1271. B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle, "Propagation Characteristics of Boolean Functions," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 161–173.

1272. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.

1273. W. Price, "Key Management for Data Encipherment," *Security: Proceedings of IFIP/SEC '83*, North Holland: Elsevier Science Publishers, 1983.

1274. G.P. Purdy, "A High-Security Log-in Procedure," *Communications of the ACM*, v. 17, n. 8, Aug 1974, pp. 442–445.

1275. J.-J. Quisquater, "Announcing the Smart Card with RSA Capability," *Proceedings of the Conference: IC Cards and Applications, Today and Tomorrow*, Amsterdam, 1989.

1276. J.-J. Quisquater and C. Couvreur, "Fast Decipherment Algorithm for RSA Public-Key Cryptosystem," *Electronic Letters*, v. 18, 1982, pp. 155–168.

1277. J.-J. Quisquater and J.-P. Delescaille, "Other Cycling Tests for DES," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 255–256.

1278. J.-J. Quisquater and Y.G. Desmedt, "Chinese Lotto as an Exhaustive Code-Breaking Machine," *Computer*, v. 24, n. 11, Nov 1991, pp. 14–22.

1279. J.-J. Quisquater and M. Girault, "$2n$-bit Hash Functions Using n-bit Symmetric Block Cipher Algorithms, *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 102–109.

1280. J.-J. Quisquater and L.C. Guillou, "Des Procédés d'Authentification Basés sur une Publication de Problèmes Complexes et Personnalisés dont les Solutions Maintenues Secrètes Constituent autant d'Accréditations," *Proceedings of SECURICOM '89: 7th Worldwide Congress on Computer and Communications Security and Protection*, Société d'Édition et d'Organisation d'Expositions Professionnelles, 1989, pp. 149–158. (In French.)

1281. J.-J., Myriam, Muriel, and Michaël Quisquater; L., Marie Annick, Gaïd, Anna, Gwenolé, and Soazig Guillou; and T. Berson, "How to Explain Zero-Knowledge Protocols to Your Children," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 628–631.

1282. M.O. Rabin, "Digital Signatures," *Foundations of Secure Communication*, New York: Academic Press, 1978, pp. 155–168.

1283. M.O. Rabin, "Digital Signatures and Public-Key Functions as Intractable as Factorization," MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212, Jan 1979.

1284. M.O. Rabin, "Probabilistic Algorithm for Testing Primality," *Journal of Number Theory*, v. 12, n. 1, Feb 1980, pp. 128–138.

1285. M.O. Rabin, "Probabilistic Algorithms in Finite Fields," *SIAM Journal on Computing*, v. 9, n. 2, May 1980, pp. 273–280.

1286. M.O. Rabin, "How to Exchange Secrets by Oblivious Transfer," Technical Memo TR-81, Aiken Computer Laboratory, Harvard University, 1981.

1287. M.O. Rabin, "Fingerprinting by Random Polynomials," Technical Report TR-15-81, Center for Research in Computing Technology, Harvard University, 1981.

1288. T. Rabin and M. Ben-Or, "Verifiable Secret Sharing and Multiparty Protocols with Honest Majority," *Proceedings of the 21st ACM Symposium on the Theory of Computing*, 1989, pp. 73–85.

1289. RAND Corporation, *A Million Random Digits with 100,000 Normal Deviates*, Glencoe, IL: Free Press Publishers, 1955.

1290. T.R.N. Rao, "Cryposystems Using Algebraic Codes," *International Conference on Computer Systems and Signal Processing*, Bangalore, India, Dec 1984.

1291. T.R.N. Rao, "On Struit-Tilburg Cryptanalysis of Rao-Nam Scheme," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 458–460.

1292. T.R.N. Rao and K.H. Nam, "Private-Key Algebraic-Coded Cryptosystems," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 35–48.

1293. T.R.N. Rao and K.H. Nam, "Private-Key Algebraic-Code Encryptions," *IEEE Transactions on Information Theory*, v. 35, n. 4, Jul 1989, pp. 829–833.

1294. J.A. Reeds, "Cracking Random Number Generator," *Cryptologia*, v. 1, n. 1, Jan 1977, pp. 20–26.

1295. J.A. Reeds, "Cracking a Multiplicative Congruential Encryption Algorithm," in *Information Linkage Between Applied Mathematics and Industry*, P.C.C. Wang, ed., Academic Press, 1979, pp. 467–472.

1296. J.A. Reeds, "Solution of Challenge Cipher," *Cryptologia*, v. 3, n. 2, Apr 1979, pp. 83–95.

1297. J.A. Reeds and J.L. Manferdelli, "DES Has No Per Round Linear Factors," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 377–389.

1298. J.A. Reeds and N.J.A. Sloane, "Shift Register Synthesis (Modulo m)," *SIAM Journal on Computing*, v. 14, n. 3, Aug 1985, pp. 505–513.

1299. J.A. Reeds and P.J. Weinberger, "File Security and the UNIX Crypt Command," *AT&T Technical Journal*, v. 63, n. 8, Oct 1984, pp. 1673–1683.

1300. T. Renji, "On Finite Automaton One-Key Cryptosystems," *Fast Software Encryption*, *Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 135–148.

1301. T. Renji and C. Shihua, "A Finite Automaton Public Key Cryptosystems and Digital Signature," *Chinese Journal of Computers*, v. 8, 1985, pp. 401–409. (In Chinese.)

1302. T. Renji and C. Shihua, "Two Varieties of Finite Automaton Public Key Cryptosystems and Digital Signature," *Journal of Computer Science and Tecnology*, v. 1, 1986, pp. 9–18. (In Chinese.)

1303. T. Renji and C. Shihua, "An Implementation of Identity-based Cryptosystems and Signature Schemes by Finite Automaton Public Key Cryptosystems," *Advances in Cryptology—CHINACRYPT '92*, Bejing: Science Press, 1992, pp. 87–104. (In Chinese.)

1304. T. Renji and C. Shihua, "Note on Finite Automaton Public Key Cryptosystems," *CHINACRYPT '94*, Xidian, China, 11–15 Nov 1994, pp. 76–80.

1305. Research and Development in Advanced Communication Technologies in Europe, *RIPE Integrity Primitives: Final Report of RACE Integrity Primitives Evaluation (R1040)*, RACE, June 1992.

1306. J.M. Reyneri and E.D. Karnin, "Coin Flipping by Telephone," *IEEE Transactions on Information Theory*, v. IT-30, n. 5, Sep 1984, pp. 775–776.

1307. P. Ribenboim, *The Book of Prime Number Records*, Springer-Verlag, 1988.

1308. P. Ribenboim, *The Little Book of Big Primes*, Springer-Verlag, 1991.

1309. M. Richter, "Ein Rauschgenerator zur Gewinnung won quasi-idealen Zufallszahlen für die stochastische Simulation," Ph.D. dissertation, Aachen University of Technology, 1992. (In German.)

1310. R.F. Rieden, J.B. Snyder, R.J. Widman, and W.J. Barnard, "A Two-Chip Implementation of the RSA Public Encryption Algorithm," *Proceedings of GOMAC (Government Microcircuit Applications Conference)*, Nov 1982, pp. 24–27.

1311. H. Riesel, *Prime Numbers and Computer Methods for Factorization*, Boston: Birkhaüser, 1985.

1312. K. Rihaczek, "Data Interchange and Legal Security—Signature Surrogates," *Computers & Security*, v. 13, n. 4, Sep 1994, pp. 287–293.

1313. V. Rijmen and B. Preneel, "Improved Characteristics for Differential Cryptanalysis of Hash Functions Based on Block Ciphers," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.

1314. R.L. Rivest, "A Description of a Single-Chip Implementation of the RSA Cipher," *LAMBDA Magazine*, v. 1, n. 3, Fall 1980, pp. 14–18.

1315. R.L. Rivest, "Statistical Analysis of the Hagelin Cryptograph," *Cryptologia*, v. 5, n. 1, Jan 1981, pp. 27–32.

1316. R.L. Rivest, "A Short Report on the RSA Chip," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, p. 327.

1317. R.L. Rivest, "RSA Chips (Past/Present/Future)," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 159–168.

1318. R.L. Rivest, "The MD4 Message Digest Algorithm," RFC 1186, Oct 1990.

1319. R.L. Rivest, "The MD4 Message Digest Algorithm," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 303–311.

1320. R.L. Rivest, "The RC4 Encryption Algorithm," RSA Data Security, Inc., Mar 1992.

1321. R.L. Rivest, "The MD4 Message Digest Algorithm," RFC 1320, Apr 1992.

1322. R.L. Rivest, "The MD5 Message Digest Algorithm," RFC 1321, Apr 1992.

1323. R.L. Rivest, "Dr. Ron Rivest on the Difficulty of Factoring," *Ciphertext: The RSA Newsletter*, v. 1, n. 1, Fall 1993, pp. 6, 8.

1324. R.L. Rivest, "The RC5 Encryption Algorithm," *Dr. Dobb's Journal*, v. 20, n. 1, Jan 95, pp. 146–148.

1325. R.L. Rivest, "The RC5 Encryption Algorithm," *K.U. Leuven Workshop on Cryptographic Algorithms*, Springer-Verlag, 1995, to appear.

1326. R.L. Rivest, M.E. Hellman, J.C. Anderson, and J.W. Lyons, "Responses to NIST's Proposal," *Communications of the ACM*, v. 35, n. 7, Jul 1992, pp. 41–54.

1327. R.L. Rivest and A. Shamir, "How to Expose an Eavesdropper," *Communications of the ACM*, v. 27, n. 4, Apr 1984, pp. 393–395.

1328. R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, v. 21, n. 2, Feb 1978, pp. 120–126.

1329. R.L. Rivest, A. Shamir, and L.M. Adleman, "On Digital Signatures and Public Key

Cryptosystems," MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212, Jan 1979.

1330. R.L. Rivest, A. Shamir, and L.M. Adleman, "Cryptographic Communications System and Method," U.S. Patent #4,405,829, 20 Sep 1983.

1331. M.J.B. Robshaw, "Implementations of the Search for Pseudo-Collisions in MD5," Technical Report TR-103, Version 2.0, RSA Laboratories, Nov 1993.

1332. M.J.B. Robshaw, "The Final Report of RACE 1040: A Technical Summary," Technical Report TR-9001, Version 1.0, RSA Laboratories, Jul 1993.

1333. M.J.B. Robshaw, "On Evaluating the Linear Complexity of a Sequence of Least Period 2^n," *Designs, Codes and Cryptography,* v. 4, n. 3, 1994, pp. 263–269.

1334. M.J.B. Robshaw, "Block Ciphers," Technical Report TR-601, RSA Laboratories, Jul 1994.

1335. M.J.B. Robshaw, "MD2, MD4, MD5, SHA, and Other Hash Functions," Technical Report TR-101, Version 3.0, RSA Laboratories, Jul 1994.

1336. M.J.B. Robshaw, "On Pseudo-Collisions in MD5," Technical Report TR-102, Version 1.1, RSA Laboratories, Jul 1994.

1337. M.J.B. Robshaw, "Security of RC4," Technical Report TR-401, RSA Laboratories, Jul 1994.

1338. M.J.B. Robshaw, personal communication, 1995.

1339. M. Roe, "Reverse Engineering of an EES Device," *K.U. Leuven Workshop on Cryptographic Algorithms,* Springer-Verlag, 1995, to appear.

1340. P. Rogaway and D. Coppersmith, "A Software-Oriented Encryption Algorithm," *Fast Software Encryption, Cambridge Security Workshop Proceedings,* Springer-Verlag, 1994, pp. 56–63.

1341. H.L. Rogers, "An Overview of the Candware Program," *Proceedings of the 3rd Annual Symposium on Physical/Electronic Security,* Armed Forces Communications and Electronics Association, paper 31, Aug 1987.

1342. J. Rompel, "One-Way Functions Are Necessary and Sufficient for Secure Signatures," *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing,* 1990, pp. 387–394.

1343. T. Rosati, "A High Speed Data Encryption Processor for Public Key Cryptography," *Proceedings of the IEEE Custom Integrated Circuits Conference,* 1989, pp. 12.3.1–12.3.5.

1344. O.S. Rothaus, "On 'Bent' Functions," *Journal of Combinational Theory,* Series A, v. 20, n. 3, 1976, pp. 300–305.

1345. RSA Laboratories, "PKCS #1: RSA Encryption Standard," version 1.5, Nov 1993.

1346. RSA Laboratories, "PKCS #3: Diffie-Hellman Key-Agreement Standard," version 1.4, Nov 1993.

1347. RSA Laboratories, "PKCS #5: Password-Based Encryption Standard," version 1.5, Nov 1993.

1348. RSA Laboratories, "PKCS #6: Extended-Certificate Syntax Standard," version 1.5, Nov 1993.

1349. RSA Laboratories, "PKCS #7: Cryptographic Message Syntax Standard," version 1.5, Nov 1993.

1350. RSA Laboratories, "PKCS #8: Private Key Information Syntax Standard," version 1.2, Nov 1993.

1351. RSA Laboratories, "PKCS #9: Selected Attribute Types," version 1.1, Nov 1993.

1352. RSA Laboratories, "PKCS #10: Certification Request Syntax Standard," version 1.0, Nov 1993.

1353. RSA Laboratories, "PKCS #11: Cryptographic Token Interface Standard," version 1.0, Apr 95.

1354. RSA Laboratories, "PKCS #12: Public Key User Information Syntax Standard," version 1.0, 1995.

1355. A.D. Rubin and P. Honeyman, "Formal Methods for the Analysis of Authentication Protocols," draft manuscript, 1994.

1356. F. Rubin, "Decrypting a Stream Cipher Based on J-K Flip-Flops," *IEEE Transactions on Computing,* v. C-28, n. 7, Jul 1979, pp. 483–487.

1357. R.A. Rueppel, *Analysis and Design of Stream Ciphers,* Springer-Verlag, 1986.

1358. R.A. Rueppel, "Correlation Immunity and the Summation Combiner," *Advances in Cryptology—EUROCRYPT '85,* Springer-Verlag, 1986, pp. 260–272.

1359. R.A. Rueppel, "When Shift Registers Clock Themselves," *Advances in Cryptology—EUROCRYPT '87 Proceedings,* Springer-Verlag, 1987, pp. 53–64.

1360. R.A. Rueppel, "Security Models and Notions for Stream Ciphers," *Cryptogra-*

phy and Coding II, C. Mitchell, ed., Oxford: Clarendon Press, 1992, pp. 213–230.

1361. R.A. Rueppel, "On the Security of Schnorr's Pseudo-Random Sequence Generator," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 423–428.

1362. R.A. Rueppel, "Stream Ciphers," *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, pp. 65–134.

1363. R.A. Rueppel and J.L. Massey, "The Knapsack as a Nonlinear Function," *IEEE International Symposium on Information Theory*, Brighton, UK, May 1985.

1364. R.A. Rueppel and O.J. Staffelbach, "Products of Linear Recurring Sequences with Maximum Complexity," *IEEE Transactions on Information Theory*, v. IT-33, n. 1, Jan 1987, pp. 124–131.

1365. D. Russell and G.T. Gangemi, *Computer Security Basics*, O'Reilly and Associates, Inc., 1991.

1366. S. Russell and P. Craig, "Privacy Enhanced Mail Modules for ELM," *Proceedings of the Internet Society 1994 Workshop on Network and Distributed System Security*, The Internet Society, 1994, pp. 21–34.

1367. D.F.H. Sadok and J. Kelner, "Privacy Enhanced Mail Design and Implementation Perspectives," *Computer Communications Review*, v. 24, n. 3, Jul 1994, pp. 38–46.

1368. K. Sakano, "Digital Signatures with User-Flexible Reliability," *Proceedings of the 1993 Symposium on Cryptography and Information Security (SCIS 93)*, Shuzenji, Japan, 28–30 Jan 1993, pp. 5C.1–8.

1369. K. Sakano, C. Park, and K. Kurosawa, "(*k,n*) Threshold Undeniable Signature Scheme," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 184–193.

1370. K. Sako, "Electronic Voting Schemes Allowing Open Objection to the Tally," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. E77-A, n. 1, 1994, pp. 24–30.

1371. K. Sako and J. Kilian, "Secure Voting Using Partially Compatible Homomorphisms," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994, p. 411–424.

1372. K. Sako and J. Kilian, "Receipt-Free Mix-Type Voting Scheme—A Practical Solution to the Implementation of a Voting Booth," *Advances in Cryptology—EUROCRYPT '95 Proceedings*, Springer-Verlag, 1995, pp. 393–403.

1373. A. Salomaa, *Public-Key Cryptography*, Springer-Verlag, 1990.

1374. A. Salomaa and L. Santean, "Secret Selling of Secrets with Many Buyers," *ETACS Bulletin*, v. 42, 1990, pp. 178–186.

1375. M. Sántha and U.V. Vazirani, "Generating Quasi-Random Sequences from Slightly Random Sources," *Proceedings of the 25th Annual Symposium on the Foundations of Computer Science*, 1984, pp. 434–440.

1376. M. Sántha and U.V. Vazirani, "Generating Quasi-Random Sequences from Slightly Random Sources," *Journal of Computer and System Sciences*, v. 33, 1986, pp. 75–87.

1377. S. Saryazdi, "An Extension to ElGamal Public Key Cryptosystem with a New Signature Scheme," *Proceedings of the 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing*, North Holland: Elsevier Science Publishers, 1990, pp. 195–198.

1378. J.E. Savage, "Some Simple Self-Synchronizing Digital Data Scramblers," *Bell System Technical Journal*, v. 46, n. 2, Feb 1967, pp. 448–487.

1379. B.P. Schanning, "Applying Public Key Distribution to Local Area Networks," *Computers & Security*, v. 1, n. 3, Nov 1982, pp. 268–274.

1380. B.P. Schanning, S.A. Powers, and J. Kowalchuk, "MEMO: Privacy and Authentication for the Automated Office," *Proceedings of the 5th Conference on Local Computer Networks*, IEEE Press, 1980, pp. 21–30.

1381. Schaumuller-Bichl, "Zur Analyse des Data Encryption Standard und Synthese Verwandter Chiffriersysteme," Ph.D. dissertation, Linz University, May 1981. (In German.)

1382. Schaumuller-Bichl, "On the Design and Analysis of New Cipher Systems Related to the DES," Technical Report, Linz University, 1983.

1383. A. Scherbius, "Ciphering Machine," U.S. Patent #1,657,411, 24 Jan 1928.

1384. J.I. Schiller, "Secure Distributed Computing," *Scientific American*, v. 271, n. 5, Nov 1994, pp. 72–76.

1385. R. Schlafly, "Complaint Against Exclusive Federal Patent License," Civil Action File No. C-93 20450, United States District Court for the Northern District of California.

1386. B. Schneier, "One-Way Hash Functions," *Dr. Dobb's Journal*, v. 16, n. 9, Sep 1991, pp. 148–151.

1387. B. Schneier, "Data Guardians," *MacWorld*, v. 10, n. 2, Feb 1993, pp. 145–151.

1388. B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 191–204.

1389. B. Schneier, "The Blowfish Encryption Algorithm," *Dr. Dobb's Journal*, v. 19, n. 4, Apr 1994, pp. 38–40.

1390. B. Schneier, *Protect Your Macintosh*, Peachpit Press, 1994.

1391. B. Schneier, "Designing Encryption Algorithms for Real People," *Proceedings of the 1994 ACM SIGSAC New Security Paradigms Workshop*, IEEE Computer Society Press, 1994, pp. 63–71.

1392. B. Schneier, "A Primer on Authentication and Digital Signatures," *Computer Security Journal*, v. 10, n. 2, 1994, pp. 38–40.

1393. B. Schneier, "The GOST Encryption Algorithm," *Dr. Dobb's Journal*, v. 20, n. 1, Jan 95, pp. 123–124.

1394. B. Schneier, *E-Mail Security* (with PGP and PEM) New York: John Wiley & Sons, 1995.

1395. C.P. Schnorr, "On the Construction of Random Number Generators and Random Function Generators," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, pp. 225–232.

1396. C.P. Schnorr, "Efficient Signature Generation for Smart Cards," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 239–252.

1397. C.P. Schnorr, "Efficient Signature Generation for Smart Cards," *Journal of Cryptology*, v. 4, n. 3, 1991, pp. 161–174.

1398. C.P. Schnorr, "Method for Identifying Subscribers and for Generating and Verifying Electronic Signatures in a Data Exchange System," U.S. Patent #4,995,082, 19 Feb 1991.

1399. C.P. Schnorr, "An Efficient Cryptographic Hash Function," presented at the rump session of CRYPTO '91, Aug 1991.

1400. C.P. Schnorr, "FFT-Hash II, Efficient Cryptographic Hashing," *Advances in Cryptology—EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 45–54.

1401. C.P. Schnorr and W. Alexi, "RSA-bits are $0.5 + \varepsilon$ Secure," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 113–126.

1402. C.P. Schnorr and S. Vaudenay, "Parallel FFT-Hashing," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 149–156.

1403. C.P. Schnorr and S. Vaudenay, "Black Box Cryptanalysis of Hash Networks Based on Multipermutations," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.

1404. W. Schwartau, *Information Warfare: Chaos on the Electronic Superhighway*, New York: Thunders Mouth Press, 1994.

1405. R. Scott, "Wide Open Encryption Design Offers Flexible Implementations," *Cryptologia*, v. 9, n. 1, Jan 1985, pp. 75–90.

1406. J. Seberry, "A Subliminal Channel in Codes for Authentication without Secrecy," *Ars Combinatorica*, v. 19A, 1985, pp. 337–342.

1407. J. Seberry and J. Pieprzyk, *Cryptography: An Introduction to Computer Security*, Englewood Cliffs, N.J.: Prentice-Hall, 1989.

1408. J. Seberry, X.-M. Zhang, and Y. Zheng, "Nonlinearly Balanced Boolean Functions and Their Propagation Characteristics," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1994, pp. 49–60.

1409. H. Sedlack, "The RSA Cryptography Processor: The First High Speed One-Chip Solution," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 95–105.

1410. H. Sedlack and U. Golze, "An RSA Cryptography Processor," *Microprocessing and Microprogramming*, v. 18, 1986, pp. 583–590.

1411. E.S. Selmer, *Linear Recurrence over Finite Field*, University of Bergen, Norway, 1966.

1412. J.O. Shallit, "On the Worst Case of Three Algorithms for Computing the Jacobi Symbol," *Journal of Symbolic Computation*, v. 10, n. 6, Dec 1990, pp. 593–610.
1413. A. Shamir, "A Fast Signature Scheme," MIT Laboratory for Computer Science, Technical Memorandum, MIT/LCS/TM-107, Massachusetts Institute of Technology, Jul 1978.
1414. A. Shamir, "How to Share a Secret," *Communications of the ACM*, v. 24, n. 11, Nov 1979, pp. 612–613.
1415. A. Shamir, "On the Cryptocomplexity of Knapsack Systems," *Proceedings of the 11th ACM Symposium on the Theory of Computing*, 1979, pp. 118–129.
1416. A. Shamir, "The Cryptographic Security of Compact Knapsacks," MIT Library for Computer Science, Technical Memorandum, MIT/LCS/TM-164, Massachusetts Institute of Technology, 1980.
1417. A. Shamir, "On the Generation of Cryptographically Strong Pseudo-Random Sequences," *Lecture Notes in Computer Science 62: 8th International Colloquium on Automata, Languages, and Programming*, Springer-Verlag, 1981.
1418. A. Shamir, "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press, 1983, pp. 279–288.
1419. A. Shamir, "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," *Proceedings of the 23rd IEEE Symposium on the Foundations of Computer Science*, 1982, pp. 145–152.
1420. A. Shamir, "On the Generation of Cryptographically Strong Pseudo-Random Sequences," *ACM Transactions on Computer Systems*, v. 1, n. 1, Feb 1983, pp. 38–44.
1421. A. Shamir, "A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem," *IEEE Transactions on Information Theory*, v. IT-30, n. 5, Sep 1984, pp. 699–704.
1422. A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," *Advances in Cryptology: Proceedings of CRYPTO 84*, Springer-Verlag, 1985, pp. 47–53.
1423. A. Shamir, "On the Security of DES," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 280–281.
1424. A. Shamir, lecture at SECURICOM '89.
1425. A. Shamir, "Efficient Signature Schemes Based on Birational Permutations," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 1–12.
1426. A. Shamir, personal communication, 1993.
1427. A. Shamir and A. Fiat, "Method, Apparatus and Article for Identification and Signature," U.S. Patent #4,748,668, 31 May 1988.
1428. A. Shamir and R. Zippel, "On the Security of the Merkle-Hellman Cryptographic Scheme," *IEEE Transactions on Information Theory*, v. 26, n. 3, May 1980, pp. 339–340.
1429. M. Shand, P. Bertin, and J. Vuillemin, "Hardware Speedups in Long Integer Multiplication," *Proceedings of the 2nd Annual ACM Symposium on Parallel Algorithms and Architectures*, 1990, pp. 138–145.
1430. D. Shanks, *Solved and Unsolved Problems in Number Theory*, Washington D.C.: Spartan, 1962.
1431. C.E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, v. 27, n. 4, 1948, pp. 379–423, 623–656.
1432. C.E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, v. 28, n. 4, 1949, pp. 656–715.
1433. C.E. Shannon, *Collected Papers: Claude Elmwood Shannon*, N.J.A. Sloane and A.D. Wyner, eds., New York: IEEE Press, 1993.
1434. C.E. Shannon, "Predication and Entropy in Printed English," *Bell System Technical Journal*, v. 30, n. 1, 1951, pp. 50–64.
1435. A. Shimizu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," *Transactions of IEICE of Japan*, v. J70-D, n. 7, Jul 87, pp. 1413–1423. (In Japanese.)
1436. A. Shimizu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 267–278.
1437. A. Shimizu and S. Miyaguchi, "FEAL—Fast Data Encipherment Algorithm," *Systems and Computers in Japan*, v. 19, n. 7, 1988, pp. 20–34, 104–106.
1438. A. Shimizu and S. Miyaguchi, "Data Randomization Equipment," U.S. Patent #4,850,019, 18 Jul 1989.

1439. M. Shimada, "Another Practical Public-key Cryptosystem," *Electronics Letters*, v. 28, n. 23, 5 Nov 1992, pp. 2146–2147.

1440. K. Shirriff, personal communication, 1993.

1441. H. Shizuya, T. Itoh, and K. Sakurai, "On the Complexity of Hyperelliptic Discrete Logarithm Problem," *Advances in Cryptology—EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 337–351.

1442. Z. Shmuley, "Composite Diffie-Hellman Public-Key Generating Systems Are Hard to Break," Computer Science Department, Technion, Haifa, Israel, Technical Report 356, Feb 1985.

1443. P.W. Shor, "Algorithms for Quantum Computation: Discrete Log and Factoring," *Proceedings of the 35th Symposium on Foundations of Computer Science*, 1994, pp. 124–134.

1444. L. Shroyer, letter to NIST regarding DSS, 17 Feb 1992.

1445. C. Shu, T. Matsumoto, and H. Imai, "A Multi-Purpose Proof System, *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. E75-A, n. 6, Jun 1992, pp. 735–743.

1446. E.H. Sibley, "Random Number Generators: Good Ones Are Hard to Find," *Communications of the ACM*, v. 31, n. 10, Oct 1988, pp. 1192–1201.

1447. V.M. Sidenikov and S.O. Shestakov, "On Encryption Based on Generalized Reed-Solomon Codes," *Diskretnaya Math*, v. 4, 1992, pp. 57–63. (In Russian.)

1448. V.M. Sidenikov and S.O. Shestakov, "On Insecurity of Cryptosystems Based on Generalized Reed-Solomon Codes," unpublished manuscript, 1992.

1449. D.P. Sidhu, "Authentication Protocols for Computer Networks," *Computer Networks and ISDN Systems*, v. 11, n. 4, Apr 1986, pp. 297–310.

1450. T. Siegenthaler, "Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications," *IEEE Transactions on Information Theory*, v. IT-30, n. 5, Sep 1984, pp. 776–780.

1451. T. Siegenthaler, "Decrypting a Class of Stream Ciphers Using Ciphertext Only," *IEEE Transactions on Computing*, v. C-34, Jan 1985, pp. 81–85.

1452. T. Siegenthaler, "Cryptanalyst's Representation of Nonlinearity Filtered *ml*-sequences," *Advances in Cryptology—EUROCRYPT '85*, Springer-Verlag, 1986, pp. 103–110.

1453. R.D. Silverman, "The Multiple Polynomial Quadratic Sieve," *Mathematics of Computation*, v. 48, n. 177, Jan 1987, pp. 329–339.

1454. G.J. Simmons, "Authentication without Secrecy: A Secure Communication Problem Uniquely Solvable by Asymmetric Encryption Techniques," *Proceedings of IEEE EASCON '79*, 1979, pp. 661–662.

1455. G.J. Simmons, "Some Number Theoretic Questions Arising in Asymmetric Encryption Techniques," *Annual Meeting of the American Mathematical Society*, AMS Abstract 763.94.1, 1979, pp. 136–151.

1456. G.J. Simmons, "High Speed Arithmetic Using Redundant Number Systems," *Proceedings of the National Telecommunications Conference*, 1980, pp. 49.3.1–49.3.2.

1457. G.J. Simmons, "A 'Weak' Privacy Protocol Using the RSA Cryptosystem," *Cryptologia*, v. 7, n. 2, Apr 1983, pp. 180–182.

1458. G.J. Simmons, "The Prisoner's Problem and the Subliminal Channel," *Advances in Cryptology: Proceedings of CRYPTO '83*, Plenum Press, 1984, pp. 51–67.

1459. G.J. Simmons, "The Subliminal Channel and Digital Signatures," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 364–378.

1460. G.J. Simmons, "A Secure Subliminal Channel (?)," *Advances in Cryptology—CRYPTO '85 Proceedings*, Springer-Verlag, 1986, pp. 33–41.

1461. G.J. Simmons, "Cryptology," *Encyclopedia Britannica*, 16th edition, 1986, pp. 913–924B.

1462. G.J. Simmons, "How to (Really) Share a Secret," *Advances in Cryptology—CRYPTO '88 Proceedings*, Springer-Verlag, 1990, pp. 390–448.

1463. G.J. Simmons, "Prepositioned Secret Sharing Schemes and/or Shared Control Schemes," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 436–467.

1464. G.J. Simmons, "Geometric Shares Secret and/or Shared Control Schemes," *Advances in Cryptology—CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 216–241.

1465. G.J. Simmons, ed., *Contemporary Cryptology: The Science of Information Integrity*, IEEE Press, 1992.

1466. G.J. Simmons, "An Introduction to Shared Secret and/or Shared Control Schemes and Their Application," in *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, pp. 441–497.

1467. G.J. Simmons, "How to Insure that Data Acquired to Verify Treaty Compliance Are Trustworthy," in *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, pp. 615–630.

1468. G.J. Simmons, "The Subliminal Channels of the U.S. Digital Signature Algorithm (DSA)," *Proceedings of the Third Symposium on: State and Progress of Research in Cryptography*, Rome: Fondazone Ugo Bordoni, 1993, pp. 35–54.

1469. G.J. Simmons, "Subliminal Communication is Easy Using the DSA," *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 218–232.

1470. G.J. Simmons, "An Introduction to the Mathematics of Trust in Security Protocols," *Proceedings: Computer Security Foundations Workshop VI*, IEEE Computer Society Press, 1993, pp. 121–127.

1471. G.J. Simmons, "Protocols that Ensure Fairness," *Codes and Ciphers*, Institute of Mathematics and its Applications, 1995, pp. 383–394.

1472. G.J. Simmons, "Cryptanalysis and Protocol Failures," *Communications of the ACM*, v. 37, n. 11, Nov 1994, pp. 56–65.

1473. G.J. Simmons, "Subliminal Channels: Past and Present," *European Transactions on Telecommuncations*, v. 4, n. 4, Jul/Aug 1994, pp. 459–473.

1474. G.J. Simmons and M.J. Norris, *How to Cipher Fast Using Redundant Number Systems*, SAND-80-1886, Sandia National Laboratories, Aug 1980.

1475. A. Sinkov, *Elementary Cryptanalysis*, Mathematical Association of America, 1966.

1476. R. Siromoney and L. Matthew, "A Public Key Cryptosystem Based on Lyndon Words," *Information Processing Letters*, v. 35, n. 1, 15 Jun 1990, pp. 33–36.

1477. B. Smeets, "A Note on Sequences Generated by Clock-Controlled Shift Registers," *Advances in Cryptology—EUROCRYPT '85*, Springer-Verlag, 1986, pp. 40–42.

1478. M.E. Smid, "A Key Notarization System for Computer Networks," NBS Special Report 500-54, U.S. Department of Commerce, Oct 1979.

1479. M.E. Smid, "The DSS and the SHS," *Federal Digital Signature Applications Symposium*, Rockville, MD, 17–18 Feb 1993.

1480. M.E. Smid and D.K. Branstad, "The Data Encryption Standard: Past and Future," *Proceedings of the IEEE*, v. 76, n. 5., May 1988, pp. 550–559.

1481. M.E. Smid and D.K. Branstad, "The Data Encryption Standard: Past and Future," in *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, ed., IEEE Press, 1992, pp. 43–64.

1482. J.L. Smith, "The Design of Lucifer, A Cryptographic Device for Data Communications," IBM Research Report RC3326, 1971.

1483. J.L. Smith, "Recirculating Block Cipher Cryptographic System," U.S. Patent #3,796,830, 12 Mar 1974.

1484. J.L. Smith, W.A. Notz, and P.R. Osseck, "An Experimental Application of Cryptography to a Remotely Accessed Data System," *Proceedings of the ACM Annual Conference*, Aug 1972, pp. 282–290.

1485. K. Smith, "Watch Out Hackers, Public Encryption Chips Are Coming," *Electronics Week*, 20 May 1985, pp. 30–31.

1486. P. Smith, "LUC Public-Key Encryption," *Dr. Dobb's Journal*, v. 18, n. 1, Jan 1993, pp. 44–49.

1487. P. Smith and M. Lennon, "LUC: A New Public Key System," *Proceedings of the Ninth International Conference on Information Security, IFIP/Sec 1993*, North Holland: Elsevier Science Publishers, 1993, pp. 91–111.

1488. E. Snekkenes, "Exploring the BAN Approach to Protocol Analysis," *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 1991, pp. 171–181.

1489. B. Snow, "Multiple Independent Binary Bit Stream Generator," U.S. Patent #5,237,615, 17 Aug 1993.

1490. R. Solovay and V. Strassen, "A Fast Monte-Carlo Test for Primality," *SIAM Journal on Computing*, v. 6, Mar 1977, pp. 84–85; erratum in ibid, v. 7, 1978, p. 118.

1491. T. Sorimachi, T. Tokita, and M. Matsui, "On a Cipher Evaluation Method Based on Differential Cryptanalysis," *Proceedings of the 1994 Symposium on Cryptography* *and Information Security (SCIS 94)*, Lake Biwa, Japan, 27–29 Jan 1994, pp. 4C.1–9. (In Japanese.)

1492. A. Sorkin, "Lucifer, a Cryptographic Algorithm," *Cryptologia*, v. 8, n. 1, Jan 1984, pp. 22–41.

1493. W. Stallings, "Kerberos Keeps the Ethernet Secure," *Data Communications*, Oct 1994, pp. 103–111.

1494. W. Stallings, *Network and Internetwork Security*, Englewood Cliffs, N.J.: Prentice-Hall, 1995.

1495. W. Stallings, *Protect Your Privacy: A Guide for PGP Users*, Englewood Cliffs, N.J.: Prentice-Hall, 1995.

1496. Standards Association of Australia, "Australian Standard 2805.4 1985: Electronic Funds Transfer—Requirements for Interfaces: Part 4—Message Authentication," SAA, North Sydney, NSW, 1985.

1497. Standards Association of Australia, "Australian Standard 2805.5 1985: Electronic Funds Transfer—Requirements for Interfaces: Part 5—Data Encipherment Algorithm," SAA, North Sydney, NSW, 1985.

1498. Standards Association of Australia, "Australian Standard 2805.5.3: Electronic Data Transfer—Requirements for Interfaces: Part 5.3—Data Encipherment Algorithm 2," SAA, North Sydney, NSW, 1992.

1499. J.G. Steiner, B.C. Neuman, and J.I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," *USENIX Conference Proceedings*, Feb 1988, pp. 191–202.

1500. J. Stern, "Secret Linear Congruential Generators Are Not Cryptographically Secure," *Proceedings of the 28th Symposium on Foundations of Computer Science*, 1987, pp. 421–426.

1501. J. Stern, "A New Identification Scheme Based on Syndrome Decoding," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 13–21.

1502. A. Stevens, "Hacks, Spooks, and Data Encryption," *Dr. Dobb's Journal*, v. 15, n. 9, Sep 1990, pp. 127–134, 147–149.

1503. R. Struik, "On the Rao-Nam Private-Key Cryptosystem Using Non-Linear Codes," *IEEE 1991 Symposium on Information Theory*, Budapest, Hungary, 1991.

1504. R. Struik and J. van Tilburg, "The Rao-Nam Scheme Is Insecure against a Chosen-Plaintext Attack," *Advances in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 445–457.

1505. S.G. Stubblebine and V.G. Gligor, "Protecting the Integrity of Privacy-Enhanced Mail with DES-Based Authentication Codes," *Proceedings of the Privacy and Security Research Group 1993 Workshop on Network and Distributed System Security*, The Internet Society, 1993, pp. 75–80.

1506. R. Sugarman, "On Foiling Computer Crime," *IEEE Spectrum*, v. 16, n. 7, Jul 79, pp. 31–32.

1507. H.N. Sun and T. Hwang, "Public-key ID-Based Cryptosystem," *Proceedings of the 25th Annual 1991 IEEE International Carnahan Conference on Security Technology*, Taipei, Taiwan, 1–3 Oct 1991, pp. 142–144.

1508. P.F. Syverson, "Formal Semantics for Logics of Computer Protocols," *Proceedings of the Computer Security Foundations Workshop III*, IEEE Computer Society Press, 1990, pp. 32–41.

1509. P.F. Syverson, "The Use of Logic in the Analysis of Cryptographic Protocols," *Proceedings of the 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 1991, pp. 156–170.

1510. P.F. Syverson, "Knowledge, Belief, and Semantics in the Analysis of Cryptographic Protocols," *Journal of Computer Security*, v. 1, n. 3, 1992, pp. 317–334.

1511. P.F. Syverson, "Adding Time to a Logic Authentication," *1st ACM Conference on Computer and Communications Security*, ACM Press, 1993, pp. 97–106.

1512. P.F. Syverson and C.A. Meadows, "A Logical Language for Specifying Cryptographic Protocol Requirements," *Proceedings of the 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, 1993, pp. 14–28.

1513. P.F. Syverson and C.A. Meadows, "Formal Requirements for Key Distribution Protocols," *Advances in Cryptology—EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.

1514. P.F. Syverson and P.C. van Oorschot, "On Unifying Some Cryptographic Protocol Logics," *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, 1994, pp. 165–177.

1515. H. Tanaka, "A Realization Scheme for the Identity-Based Cryptosystem," *Advances* *in Cryptology—CRYPTO '87 Proceedings*, Springer-Verlag, 1988, pp. 340–349.

1516. H. Tanaka, "A Realization Scheme for the Identity-Based Cryptosystem," *Electronics and Communications in Japan, Part 3 (Fundamental Electronic Science)*, v. 73, n. 5, May 1990, pp. 1–7.

1517. H. Tanaka, "Identity-Based Noninteractive Common-Key Generation and Its Application to Cryptosystems," *Transactions of the Institute of Electronics, Information, and Communication Engineers*, v. J75-A, n. 4, Apr 1992, pp. 796–800.

1518. J. Tardo and K. Alagappan, "SPX: Global Authentication Using Public Key Certificates," *Proceedings of the 1991 IEEE Computer Society Symposium on Security and Privacy*, 1991, pp. 232–244.

1519. J. Tardo, K. Alagappan, and R. Pitkin, "Public Key Based Authentication Using Internet Certificates," *USENIX Security II Workshop Proceedings*, 1990, pp. 121–123.

1520. A. Tardy-Corfdir and H. Gilbert, "A Known Plaintext Attack of FEAL-4 and FEAL-6," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 172–182.

1521. M. Tatebayashi, N. Matsuzaki, and D.B. Newman, "Key Distribution Protocol for Digital Mobile Communication System," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 324–333.

1522. M. Taylor, "Implementing Privacy Enhanced Mail on VMS," *Proceedings of the Privacy and Security Research Group 1993 Workshop on Network and Distributed System Security*, The Internet Society, 1993, pp. 63–68.

1523. R. Taylor, "An Integrity Check Value Algorithm for Stream Ciphers," *Advances in Cryptology—CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 40–48.

1524. T. Tedrick, "Fair Exchange of Secrets," *Advances in Cryptology: Proceedings of CRYPTO '84*, Springer-Verlag, 1985, pp. 434–438.

1525. R. Terada and P.G. Pinheiro, "How to Strengthen FEAL against Differential Cryptanalysis," *Proceedings of the 1995 Japan-Korea Workshop on Information Security and Cryptography*, Inuyama, Japan, 24–27 Jan 1995, pp. 153–162.

1526. J.-P. Tillich and G. Zémor, "Hashing with SL_2," *Advances in Cryptology—CRYPTO '94 Proceedings*, Springer-Verlag, 1994, pp. 40–49.

1527. T. Tokita, T. Sorimachi, and M. Matsui, "An Efficient Search Algorithm for the Best Expression on Linear Cryptanalysis," IEICE Japan, Technical Report, ISEC93-97, 1994.

1528. M. Tompa and H. Woll, "Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information," *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science*, 1987, pp. 472–482.

1529. M. Tompa and H. Woll, "How to Share a Secret with Cheaters," *Journal of Cryptology*, v. 1, n. 2, 1988, pp. 133–138.

1530. M.-J. Toussaint, "Verification of Cryptographic Protocols," Ph.D. dissertation, Université de Liège, 1991.

1531. M.-J. Toussaint, "Deriving the Complete Knowledge of Participants in Cryptographic Protocols," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 24–43.

1532. M.-J. Toussaint, "Separating the Specification and Implementation Phases in Cryptology," *ESORICS 92, Proceedings of the Second European Symposium on Research in Computer Security*, Springer-Verlag, 1992, pp. 77–101.

1533. P.D. Townsend, J.G. Rarity, and P.R. Tapster, "Enhanced Single Photon Fringe Visibility in a 10 km-Long Prototype Quantum Cryptography Channel," *Electronics Letters*, v. 28, n. 14, 8 Jul 1993, pp. 1291–1293.

1534. S.A. Tretter, "Properties of PN^2 Sequences," *IEEE Transactions on Information Theory*, v. IT-20, n. 2, Mar 1974, pp. 295–297.

1535. H. Truman, "Memorandum for: The Secretary of State, The Secretary of Defense," A 20707 5/4/54/OSO, NSA TS CONTL. NO 73-00405, 24 Oct 1952.

1536. Y.W. Tsai and T. Hwang, "ID Based Public Key Cryptosystem Based on Okamoto and Tanaka's ID Based One-Way Communications Scheme," *Electronics Letters*, v. 26, n. 10, 1 May 1990, pp. 666–668.

1537. G. Tsudik, "Message Authentication with One-Way Hash Functions," *ACM Computer Communications Review*, v. 22, n. 5, 1992, pp. 29–38.

1538. S. Tsujii and K. Araki, "A Rebuttal to Coppersmith's Attacking Method," memorandum presented at Crypto '94, Aug 1994.

1539. S. Tsujii, K. Araki, J. Chao, T. Sekine, and Y. Matsuzaki, "ID-Based Key Sharing Scheme—Cancellation of Random Numbers by Iterative Addition," IEICE Japan, Technical Report, ISEC 92-47, Oct 1992.

1540. S. Tsujii, K. Araki, and T. Sekine, "A New Scheme of Noninteractive ID-Based Key Sharing with Explosively High Degree of Separability," Technical Report, Department of Computer Science, Tokyo Institute of Technology, 93TR-0016, May 1993.

1541. S. Tsujii, K. Araki, and T. Sekine, "A New Scheme of Non Interactive ID-Based key Sharing with Explosively High Degree of Separability (Second Version)," Technical Report, Department of Computer Science, Tokyo Institute of Technology, 93TR-0020, Jul 1993.

1542. S. Tsujii, K. Araki, T. Sekine, and K. Tanada, "A New Scheme of Non Interactive ID-Based Key Sharing with Explosively High Degree of Separability," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 49–58.

1543. S. Tsujii, K. Araki, H. Tanaki, J. Chao, T. Sekine, and Y. Matsuzaki, "ID-Based Key Sharing Scheme—Reply to Tanaka's Comment," IEICE Japan, Technical Report, ISEC 92-60, Dec 1992.

1544. S. Tsujii and J. Chao, "A New ID-based Key Sharing System," *Advances in Cryptology—CRYPTO '91 Proceedings*, Springer-Verlag, 1992, pp. 288–299.

1545. S. Tsujii, J. Chao, and K. Araki, "A Simple ID-Based Scheme for Key Sharing," IEICE Japan, Technical Report, ISEC 92-25, Aug 1992.

1546. S. Tsujii and T. Itoh, "An ID-Based Cryptosystem Based on the Discrete Logarithm Problem," *IEEE Journal on Selected Areas in Communication*, v. 7, n. 4, May 1989, pp. 467–473.

1547. S. Tsujii and T. Itoh, "An ID-Based Cryptosystem Based on the Discrete Logarithm Problem," *Electronics Letters*, v. 23, n. 24, Nov 1989, pp. 1318–1320.

1548. S. Tsujii, K. Kurosawa, T. Itoh, A. Fujioka, and T. Matsumoto, "A Public-Key Cryptosystem Based on the Difficulty of Solving a System of Non-Linear Equations," TSUJII Laboratory Technical Memorandum, n. 1, 1986.

1549. Y. Tsunoo, E. Okamoto, and H. Doi, "Analytical Known Plain-Text Attack for FEAL-4 and Its Improvement," *Proceedings of the 1994 Symposium on Cryptography and Information Security (SCIS 93)*, 1993.

1550. Y. Tsunoo, E. Okamoto, T. Uyematsu, and M. Mambo, "Analytical Known Plain-Text Attack for FEAL-6" *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography*, Seoul, Korea, 24–26 Oct 1993, pp. 253–261.

1551. W. Tuchman, "Hellman Presents No Shortcut Solutions to DES," *IEEE Spectrum*, v. 16, n. 7, July 1979, pp. 40–41.

1552. U.S. Senate Select Committee on Intelligence, "Unclassified Summary: Involvement of NSA in the Development of the Data Encryption Standard," *IEEE Communications Magazine*, v. 16, n. 6, Nov 1978, pp. 53–55.

1553. B. Vallée, M. Girault, and P. Toffin, "How to Break Okamoto's Cryptosystem by Reducing Lattice Values," *Advances in Cryptology—EUROCRYPT '88 Proceedings*, Springer-Verlag, 1988, p. 281–291.

1554. H. Van Antwerpen, "Electronic Cash," Master's thesis, CWI, Netherlands, 1990.

1555. K. Van Espen and J. Van Mieghem, "Evaluatie en Implementatie van Authentiseringsalgoritmen," graduate thesis, ESAT Laboratorium, Katholieke Universiteit Leuven, 1989. (In Dutch.)

1556. P.C. van Oorschot, "Extending Cryptographic Logics of Belief to Key Agreement Protocols," *Proceedings of the 1st Annual ACM Conference on Computer and Communications Security*, 1993, pp. 232–243.

1557. P.C. van Oorschot, "An Alternate Explanation for Two BAN-logic 'Failures,'" *Advances in Cryptology—EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 443–447.

1558. P.C. van Oorschot and M.J. Wiener, "A Known-Plaintext Attack on Two-Key Triple Encryption," *Advances in Cryptology—EUROCRYPT '90 Proceedings*, Springer-Verlag, 1991, pp. 318–325.

1559. J. van Tilburg, "On the McEliece Cryptosystem," *Advances in Cryptology—*

CRYPTO '88 Proceedings, Springer-Verlag, 1990, pp. 119–131.

1560. J. van Tilburg, "Cryptanalysis of the Xinmei Digital Signature Scheme," *Electronics Letters*, v. 28, n. 20, 24 Sep 1992, pp. 1935–1938.
1561. J. van Tilburg, "Two Chosen-Plaintext Attacks on the Li Wang Joing Authentication and Encryption Scheme," *Applied Algebra, Algebraic Algorithms and Error Correcting Codes 10*, Springer-Verlag, 1993, pp. 332–343.
1562. J. van Tilburg, "Security-Analysis of a Class of Cryptosystems Based on Linear Error-Correcting Codes," Ph.D. dissertation, Technical University Eindhoven, 1994.
1563. A. Vandemeulebroecke, E. Vanzieleghem, T. Denayer, and P.G. Jespers, "A Single Chip 1024 Bits RSA Processor," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 219–236.
1564. J. Vanderwalle, D. Chaum, W. Fumy, C. Jansen, P. Landrock, and G. Roelofsen, "A European Call for Cryptographic Algorithms: RIPE; RACE Integrity Primitives Evaluation," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 267–271.
1565. V. Varadharajan, "Verification of Network Security Protocols," *Computers and Security*, v. 8, n. 8, Aug 1989, pp. 693–708.
1566. V. Varadharajan, "Use of a Formal Description Technique in the Specification of Authentication Protocols," *Computer Standards and Interfaces*, v. 9, 1990, pp. 203–215.
1567. S. Vaudenay, "FFT-Hash-II Is not Yet Collision-Free," *Advances in Cryptology—CRYPTO '92 Proceedings*, Springer-Verlag, pp. 587–593.
1568. S. Vaudenay, "Differential Cryptanalysis of Blowfish," unpublished manuscript, 1995.
1569. U.V. Vazirani and V.V. Vazirani, "Trapdoor Pseudo-Random Number Generators with Applications to Protocol Design," *Proceedings of the 24th IEEE Symposium on the Foundations of Computer Science*, 1983, pp. 23–30.
1570. U.V. Vazirani and V.V. Vazirani, "Efficient and Secure Pseudo-Random Number Generation," *Proceedings of the 25th IEEE Symposium on the Foundations of Computer Science*, 1984, pp. 458–463.
1571. U.V. Vazirani and V.V. Vazirani, "Efficient and Secure Pseudo-Random Number Generation," *Advances in Cryptology: Proceedings of CRYPTO '84*, Springer-Verlag, 1985, pp. 193–202.
1572. I. Verbauwhede, F. Hoornaert, J. Vanderwalle, and H. De Man, "ASIC Cryptographical Processor Based on DES," *Euro ASIC '91 Proceedings*, 1991, pp. 292–295.
1573. I. Verbauwhede, F. Hoornaert, J. Vanderwalle, H. De Man, and R. Govaerts, "Security Considerations in the Design and Implementation of a New DES Chip," *Advances in Cryptology—EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 287–300.
1574. R. Vogel, "On the Linear Complexity of Cascaded Sequences," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, Springer-Verlag, 1985, pp. 99–109.
1575. S. von Solms and D. Naccache, "On Blind Signatures and Perfect Crimes," *Computers & Security*, v. 11, 1992, pp. 581–583.
1576. V.L. Voydock and S.T. Kent, "Security Mechanisms in High-Level Networks," *ACM Computing Surveys*, v. 15, n. 2, Jun 1983, pp. 135–171.
1577. N.R. Wagner, P.S. Putter, and M.R. Cain, "Large-Scale Randomization Techniques," *Advances in Cryptology—CRYPTO '86 Proceedings*, Springer-Verlag, 1987, pp. 393–404.
1578. M. Waidner and B. Pfitzmann, "The Dining Cryptographers in the Disco: Unconditional Sender and Recipient Untraceability with Computationally Secure Serviceability," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, p. 690.
1579. S.T. Walker, "Software Key Escrow—A Better Solution for Law Enforcement's Needs?" TIS Report #533, Trusted Information Systems, Aug 1994.
1580. S.T. Walker, "Thoughts on Key Escrow Acceptability," TIS Report #534D, Trusted Information Systems, Nov 1994.
1581. S.T. Walker, S.B. Lipner, C.M. Ellison, D.K. Branstad, and D.M. Balenson, "Commercial Key Escrow—Something for Everyone—Now and for the Future," TIS Report #541, Trusted Information Systems, Jan 1995.
1582. M.Z. Wang and J.L. Massey, "The Characteristics of All Binary Sequences with Perfect Linear Complexity Profiles,"

Abstracts of Papers, EUROCRYPT '86, 20–22 May 1986.
1583. E.J. Watson, "Primitive Polynomials (Mod 2)," *Mathematics of Computation,* v. 16, 1962, p. 368.
1584. P. Wayner, "Mimic Functions," *Cryptologia,* v. 16, n. 3, Jul 1992, pp. 193–214.
1585. P. Wayner, "Mimic Functions and Tractability," draft manuscript, 1993.
1586. A.F. Webster and S.E. Tavares, "On the Design of S-Boxes," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 523–534.
1587. G. Welchman, *The Hut Six Story: Breaking the Enigma Codes,* New York: McGraw-Hill, 1982.
1588. A.L. Wells Jr., "A Polynomial Form for Logarithms Modulo a Prime," *IEEE Transactions on Information Theory,* Nov 1984, pp. 845–846.
1589. D.J. Wheeler, "A Bulk Data Encryption Algorithm," *Fast Software Encryption, Cambridge Security Workshop Proceedings,* Springer-Verlag, 1994, pp. 127–134.
1590. D.J. Wheeler, personal communication, 1994.
1591. D.J. Wheeler and R. Needham, "A Large Block DES-Like Algorithm," Technical Report 355, "Two Cryptographic Notes," Computer Laboratory, University of Cambridge, Dec 1994, pp. 1–3.
1592. D.J. Wheeler and R. Needham, "TEA, A Tiny Encryption Algorithm," Technical Report 355, "Two Cryptographic Notes," Computer Laboratory, University of Cambridge, Dec 1994, pp. 1–3.
1593. S.R. White, "Covert Distributed Processing with Computer Viruses," *Advances in Cryptology—CRYPTO '89 Proceedings,* Springer-Verlag, 1990, pp. 616–619.
1594. White House, Office of the Press Secretary, "Statement by the Press Secretary," 16 Apr 1993.
1595. B.A. Wichman and I.D. Hill, "An Efficient and Portable Pseudo-Random Number Generator," *Applied Statistics,* v. 31, 1982, pp. 188–190.
1596. M.J. Wiener, "Cryptanalysis of Short RSA Secret Exponents," *IEEE Transactions on Information Theory,* v. 36, n. 3, May 1990, pp. 553–558.
1597. M.J. Wiener, "Efficient DES Key Search," presented at the rump session of CRYPTO '93, Aug 1993.
1598. M.J. Wiener, "Efficient DES Key Search," TR-244, School of Computer Science, Carleton University, May 1994.
1599. M.V. Wilkes, *Time-Sharing Computer Systems,* New York: American Elsevier, 1968.
1600. E.A. Williams, *An Invitation to Cryptograms,* New York: Simon and Schuster, 1959.
1601. H.C. Williams, "A Modification of the RSA Public-Key Encryption Procedure," *IEEE Transactions on Information Theory,* v. IT-26, n. 6, Nov 1980, pp. 726–729.
1602. H.C. Williams, "An Overview of Factoring," *Advances in Cryptology: Proceedings of Crypto 83,* Plenum Press, 1984, pp. 71–80.
1603. H.C. Williams, "Some Public-Key Crypto-Functions as Intractable as Factorization," *Advances in Cryptology: Proceedings of CRYPTO 84,* Springer-Verlag, 1985, pp. 66–70.
1604. H.C. Williams, "Some Public-Key Crypto-Functions as Intractable as Factorization," *Cryptologia,* v. 9, n. 3, Jul 1985, pp. 223–237.
1605. H.C. Williams, "An M^3 Public-Key Encryption Scheme," *Advances in Cryptology—CRYPTO '85,* Springer-Verlag, 1986, pp. 358–368.
1606. R.S. Winternitz, "Producing One-Way Hash Functions from DES," *Advances in Cryptology: Proceedings of Crypto 83,* Plenum Press, 1984, pp. 203–207.
1607. R.S. Winternitz, "A Secure One-Way Hash Function Built from DES," *Proceedings of the 1984 Symposium on Security and Privacy,* 1984, pp. 88–90.
1608. S. Wolfram, "Random Sequence Generation by Cellular Automata," *Advances in Applied Mathematics,* v. 7, 1986, pp. 123–169.
1609. S. Wolfram, "Cryptography with Cellular Automata," *Advances in Cryptology—CRYPTO '85 Proceedings,* Springer-Verlag, 1986, pp. 429–432.
1610. T.Y.C. Woo and S.S. Lam, "Authentication for Distributed Systems," *Computer,* v. 25, n. 1, Jan 1992, pp. 39–52.
1611. T.Y.C. Woo and S.S. Lam, "'Authentication' Revisited," *Computer,* v. 25, n. 3, Mar 1992, p. 10.
1612. T.Y.C. Woo and S.S. Lam, "A Semantic Model for Authentication Protocols," *Proceedings of the 1993 IEEE Computer Soci-*

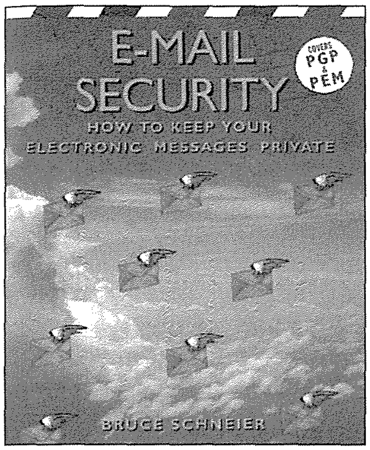
ety Symposium on Research in Security and Privacy, 1993, pp. 178–194.

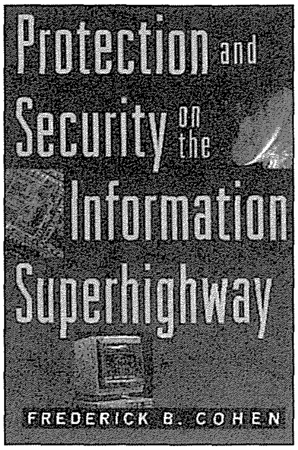
1613. M.C. Wood, technical report, Cryptech, Inc., Jamestown, NY, Jul 1990.
1614. M.C. Wood, "Method of Cryptographically Transforming Electronic Digital Data from One Form to Another," U.S. Patent #5,003,596, 26 Mar 1991.
1615. M.C. Wood, personal communication, 1993.
1616. C.K. Wu and X.M. Wang, "Determination of the True Value of the Euler Totient Function in the RSA Cryptosystem from a Set of Possibilities," *Electronics Letters,* v. 29, n. 1, 7 Jan 1993, pp. 84–85.
1617. M.C. Wunderlich, "Recent Advances in the Design and Implementation of Large Integer Factorization Algorithms," *Proceedings of 1983 Symposium on Security and Privacy,* IEEE Computer Society Press, 1983, pp. 67–71.
1618. Xerox Network System (XNS) Authentication Protocol, XSIS 098404, Xerox Corporation, Apr 1984.
1619. Y.Y. Xian, "New Public Key Distribution System," *Electronics Letters,* v. 23, n. 11, 1987, pp. 560–561.
1620. L.D. Xing and L.G. Sheng, "Cryptanalysis of New Modified Lu-Lee Cryptosystems," *Electronics Letters,* v. 26, n. 19, 13 Sep 1990, p. 1601–1602.
1621. W. Xinmei, "Digital Signature Scheme Based on Error-Correcting Codes," *Electronics Letters,* v. 26, n. 13, 21 Jun 1990, p. 898–899.
1622. S.B. Xu, D.K. He, and X.M. Wang, "An Implementation of the GSM General Data Encryption Algorithm A5," *CHINACRYPT '94,* Xidian, China, 11–15 Nov 1994, pp. 287–291. (In Chinese.)
1623. M. Yagisawa, "A New Method for Realizing Public-Key Cryptosystem," *Cryptologia,* v. 9, n. 4, Oct 1985, pp. 360–380.
1624. C.H. Yang, "Modular Arithmetic Algorithms for Smart Cards," IEICE Japan, Technical Report, ISEC92-16, 1992.
1625. C.H. Yang and H. Morita, "An Efficient Modular-Multiplication Algorithm for Smart-Card Software Implementation," IEICE Japan, Technical Report, ISEC91-58, 1991.
1626. J.H. Yang, K.C. Zeng, and Q.B. Di, "On the Construction of Large S-Boxes," *CHINACRYPT '94,* Xidian, China, 11–15 Nov 1994, pp. 24–32. (In Chinese.)
1627. A.C.-C. Yao, "Protocols for Secure Computations," *Proceedings of the 23rd IEEE Symposium on the Foundations of Computer Science,* 1982, pp. 160–164.
1628. B. Yee, "Using Secure Coprocessors," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, May 1994.
1629. S.-M. Yen, "Design and Computation of Public Key Cryptosystems," Ph.D. dissertation, National Cheng Hung University, Apr 1994.
1630. S.-M. Yen and C.-S. Lai, "New Digital Signature Scheme Based on the Discrete Logarithm," *Electronics Letters,* v. 29, n. 12, 1993, pp. 1120–1121.
1631. K. Yiu and K. Peterson, "A Single-Chip VLSI Implementation of the Discrete Exponential Public-Key Distribution System," *IBM Systems Journal,* v. 15, n. 1, 1982, pp. 102–116.
1632. K. Yiu and K. Peterson, "A Single-Chip VLSI Implementation of the Discrete Exponential Public-Key Distribution System," *Proceedings of Government Microcircuit Applications Conference,* 1982, pp. 18–23.
1633. H.Y. Youm, S.L. Lee, and M.Y. Rhee, "Practical Protocols for Electronic Cash," *Proceedings of the 1993 Korea-Japan Workshop on Information Security and Cryptography,* Seoul, Korea, 24–26 Oct 1993, pp. 10–22.
1634. M. Yung, "Cryptoprotocols: Subscriptions to a Public Key, the Secret Blocking, and the Multi-Player Mental Poker Game," *Advances in Cryptology: Proceedings of CRYPTO 84,* Springer-Verlag, 1985, 439–453.
1635. G. Yuval, "How to Swindle Rabin," *Cryptologia,* v. 3, n. 3, Jul 1979, pp. 187–190.
1636. K.C. Zeng and M. Huang, "On the Linear Syndrome Method in Cryptanalysis," *Advances in Cryptology—CRYPTO '88 Proceedings,* Springer-Verlag, 1990, pp. 469–478.
1637. K.C. Zeng, M. Huang, and T.R.N. Rao, "An Improved Linear Algorithm in Cryptanalysis with Applications," *Advances in Cryptology—CRYPTO '90 Proceedings,* Springer-Verlag, 1991, pp. 34–47.
1638. K.C. Zeng, C.-H. Yang, and T.R.N. Rao, "On the Linear Consistency Test (LCT) in Cryptanalysis with Applications," *Advances in Cryptology—CRYPTO '89*

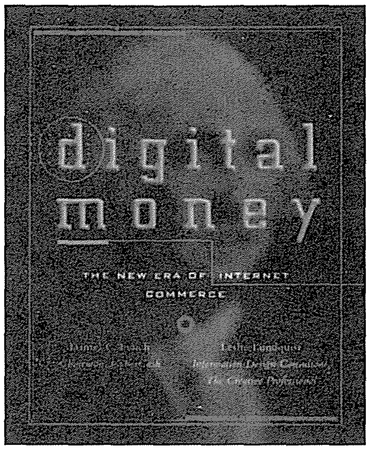
Proceedings, Springer-Verlag, 1990, pp. 164–174.

1639. K.C. Zeng, C.-H. Yang, D.-Y. Wei, and T.R.N. Rao, "Pseudorandom Bit Generators in Stream-Cipher Cryptography," *IEEE Computer*, v. 24, n. 2, Feb 1991, pp. 8–17.
1640. M. Zhang, S.E. Tavares, and L.L. Campbell, "Information Leakage of Boolean Functions and Its Relationship to Other Cryptographic Criteria," *Proceedings of the 2nd Annual ACM Conference on Computer and Communications Security*, ACM Press, 1994, pp. 156–165.
1641. M. Zhang and G. Xiao, "A Modified Design Criterion for Stream Ciphers," *CHINACRYPT '94*, Xidian, China, 11–15 Nov 1994, pp. 201–209. (In Chinese.)
1642. Y. Zheng, T. Matsumoto, and H. Imai, "Duality between two Cryptographic Primitives," *Papers of Technical Group for Information Security*, IEICE of Japan, Mar 1989, pp. 47–57.
1643. Y. Zheng, T. Matsumoto, and H. Imai, "Impossibility and Optimality Results in Constructing Pseudorandom Permutations," *Advances in Cryptology—EUROCRYPT '89 Proceedings*, Springer-Verlag, 1990, pp. 412–422.
1644. Y. Zheng, T. Matsumoto, and H. Imai, "On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses," *Advances in Cryptology—CRYPTO '89 Proceedings*, Springer-Verlag, 1990, pp. 461–480.
1645. Y. Zheng, T. Matsumoto, and H. Imai, "Duality between two Cryptographic Primitives," *Proceedings of the 8th International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Springer-Verlag, 1991, pp. 379–390.
1646. Y. Zheng, J. Pieprzyk, and J. Seberry, "HAVAL—A One-Way Hashing Algorithm with Variable Length of Output," *Advances in Crytology—AUSCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 83–104.
1647. N. Zierler, "Linear Recurring Sequences," *Journal Soc. Indust. Appl. Math.*, v. 7, n. 1, Mar 1959, pp. 31–48.
1648. N. Zierler, "Primitive Trinomials Whose Degree Is a Mersenne Exponent," *Information and Control*, v. 15, 1969, pp. 67–69.
1649. N. Zierler and J. Brillhart, "On Primitive Trinomials (mod 2)," *Information and Control*, v. 13, n. 6, Dec 1968, pp. 541–544.
1650. N. Zierler and W.H. Mills, "Products of Linear Recurring Sequences," *Journal of Algebra*, v. 27, n. 1, Oct 1973, pp. 147–157.
1651. C. Zimmer, "Perfect Gibberish," *Discover*, v. 13, n. 12, Dec 1992, pp. 92–99.
1652. P.R. Zimmermann, *The Official PGP User's Guide*, Boston: MIT Press, 1995.
1653. P.R. Zimmermann, *PGP Source Code and Internals*, Boston: MIT Press, 1995.

Index


E-MAIL SECURITY
COVERS PGP & PEM
HOW TO KEEP YOUR ELECTRONIC MESSAGES PRIVATE
BRUCE SCHNEIER


Protection and Security on the Information Superhighway
FREDERICK B. COHEN


digital money
THE NEW ERA OF INTERNET COMMERCE

The *Applied Cryptography* Source Code Disk Set

A source code disk set (three disks) associated with this book is available directly from the author. Included on these disks you will find:

Symmetric Algorithms:
- *Vigenère Cipher*
- *Playfair Cipher*
- *Hill Cipher*
- *CRYPT (1)*
- *CRYPT (3)*
- *Enigma*
- *DES - 10 versions*
- *Lucifer - 2 versions*
- *NewDES*
- *FEAL-N*
- *FEAL-XN*
- *REDOC II*
- *REDOC III*
- *LOKI89*
- *LOKI91*
- *Khufu*
- *IDEA - 3 versions*
- *CA 1.1*
- *MDC*
- *GOST*
- *BLOWFISH*
- *3-Way*
- *SAFER K-64*
- *SAFER K-128*
- *NewDE*
- *NSEA*
- *RC4*
- *PKZIP*
- *SEAL*
- *WAKE*

Public-Key Algorithms:
- *RSA*
- *Diffie-Hellman*
- *DSA*

One-Way Hash Functions:
- *Snefru*
- *N-Hash*
- *MD4 - 3 versions*
- *MD5 - 2 versions*
- *MD2*
- *SHA*
- *HAVAL*
- *RIPE-MD*

Complete Systems:
- *RIPEM*
- *PGP*
- *TIS-PEM*
- *RSAREF*

Other:
- *LaGrange Threshold Scheme*
- *Mimic Functions*
- *Probabilistic Prime Number Generation*
- *Random Number Generation using Oscillators*
- *Random Number Generation using Keyboard Latency*
- *Frequency Analysis*
- *WordPerfect Password Cracker*

Text:
- *Defense Trade Regulations*
- *DoD Orange Book*
- *European Computer Security Green Book*
- *Various NIST FIPS*
- *Various Internet RFCs*

And more!

The disks also include a file containing corrections for all mistakes found in the book, as well as any updated information on any of the topics covered in the text: new algorithms, new protocols, new cryptanalytic results, and so on.

The MS-DOS disks are available from the author, and will be updated twice a year. Cost is $40 for a set, and $120 for a two-year subscription. Please send check or money order in U.S. funds, drawn on a U.S. bank, to:

Bruce Schneier
Counterpane Systems
7115 W. North Ave., Suite 16
Oak Park, IL 60302-1002

Please allow four weeks for delivery, and include your e-mail address if you have one. Due to the export restrictions on many of the algorithms on these disks, they will only be mailed to addresses within the United States and Canada. Apologies to the foreign readers of this book.

"... the best introduction to cryptography I've ever seen .... The book the National Security Agency wanted never to be published...."

—*Wired* magazine

"... monumental ... fascinating ... comprehensive ... the definitive work on cryptography for computer programmers..."

—*Dr. Dobb's Journal*

"... easily ranks as one of the most authoritative in its field."

—*PC Magazine*

This cryptography classic provides you with a comprehensive survey of modern cryptography. The book details how programmers and electronic communications professionals can use cryptography—the technique of enciphering and deciphering messages—to maintain the privacy of computer data. It describes dozens of cryptography algorithms, gives practical advice on how to implement them into cryptographic software, and shows how they can be used to solve security problems. Covering developments in practical cryptographic techniques, this seminal work shows programmers who design computer applications, networks, and storage systems how they can build security into their software and systems.

- **Encryption algorithms, including algorithms from the former Soviet Union and South Africa, and the RC4 stream cipher**
- **Protocols for digital signatures, authentication, secure elections, digital cash, and more**

BRUCE SCHNEIER is an internationally renowned security technologist, called a "security guru" by *The Economist*. He is the author of twelve books—including *Secrets & Lies: Digital Security in a Networked World* which has become a classic as well as hundreds of articles, essays, and academic papers. His influential newsletter "Crypto-Gram" and blog "Schneier on Security" are read by over 250,000 people. Schneier is a fellow at the Berkman Center for Internet and Society at Harvard Law School, a program fellow at the New America Foundation's Open Technology Institute, a board member of the Electronic Frontier Foundation, and an Advisory Board member of the Electronic Privacy Information Center. He is also the Chief Technology Officer of Resilient Systems, Inc. You can read his blog, essays, and academic papers at www.schneier.com. He tweets at @schneierblog.

Cover Design: Wiley
Cover Painting: Mona Mark

WILEY

Programming/Security
$60.00 US/$72.00 CAN

ISBN 978-0-471-11709-4
56000