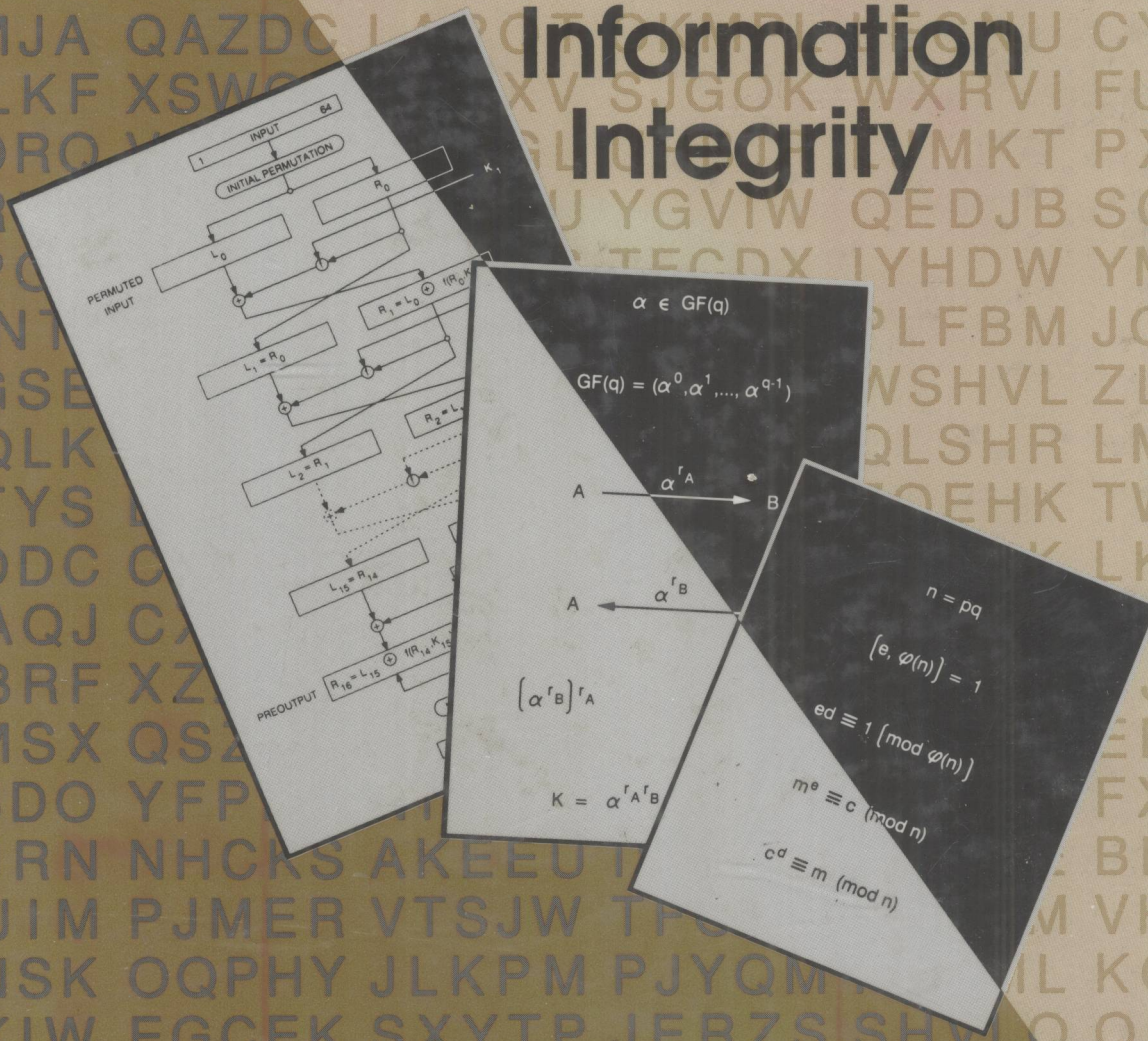
CONTEMPORARY CRYPTOLOGY

The Science of Information Integrity



EDITED BY GUSTAVUS J. SIMMONS

IEEE PRESS

Contemporary Cryptology

The Science of Information Integrity

Edited by

Gustavus J. Simmons

Sandia National Laboratories

IEEE PRESS

The Institute of Electrical and Electronics Engineers, Inc., New York

IEEE PRESS
445 Hoes Lane, PO Box 1331
Piscataway, NJ 08855-1331

Printed in the United States of America
10 9 8 7 6 5 4 3 2

ISBN 0-87942-277-7
IEEE Order Number: PC0271-7

Library of Congress Cataloging-in-Publication Data
Contemporary cryptology : the science of information integrity / edited by Gustavus J. Simmons.
p. cm.
Includes bibliographical references and index.
ISBN 0–87942–277–7
1. Computer security. 2. Telecommunication systems—Security measures. 3. Cryptography. I. Simmons, Gustavus J.
QA76.9.A25C6678 1992
005.8′2—dc20

91–19684
CIP

Contemporary Cryptology
A Foreword

Cryptology (from the Greek kryptós, "hidden," and lógos, "word") has come to be understood to be the science of secure (often interpreted to mean secret) communications. Although secrecy is certainly an important element to the security or integrity of information, it is only one element, as demonstrated by the contributing authors of this book. Information integrity is also concerned with questions of authenticity, authority, concurrence, timeliness, etc., as well as with all the problems normally addressed by documentary records. The intent of the editor and of the authors in putting this book together was to treat the subject of information integrity as comprehensively as possible—with special emphasis on those questions of information integrity whose resolution is primarily cryptographic in nature.

As the most casual reader of the technical literature, or even of the popular press, must be aware, an enormous amount of public activity in the field of cryptology has occurred during the past decade and a half. This has been marked by the appearance of several fundamental new ideas such as two-key (also public key or asymmetric) cryptography, provably secure protocols whose security is derived from mathematical problems of classifiable complexity, interactive proof systems and zero-knowledge protocols, etc., and, of course, by the widespread recognition of an urgent need for means to provide for the integrity of information in all phases of our information-intensive society [1]. This perceived need is the driving force responsible for much of the public activity.

The conduct of commerce, affairs of state, military actions, and personal affairs all depend on the parties to a transaction having confidence in there being means of accomplishing such functions as privacy, proof of identity, authority, ownership, license, signature, witnessing or notarization, date of action, certification of origination

and/or receipt, etc. As a result, an elaborate, and legally accepted, collection of procedural and physical protocols have evolved that specify how to create records (information in documentary form) in such a way that later disputes as to who is liable, or of the nature of that liability, or of when a liability was incurred, etc., can be arbitrated by a third party (typically in a court of law). The essential point is that existing precedent depends on information having a physical existence in the form of a document which may have been signed, witnessed, notarized, recorded, dated, etc.

The "proof" process, if it must be invoked, depends almost entirely on the physical instrument(s) as the means for establishing the integrity of the recorded information. In an information-intensive society however, in which the possession, control, transfer, or access to real assets is frequently based on incorporeal information—that is, information whose existence is not essentially linked to any physical record, and in which a license (to use, modify, copy, etc., valuable or sensitive information) is similarly determined, it is essential that means be found to carry out all of the functions associated with establishing the integrity of information mentioned above based only on the internal evidence present in the information itself, since this is the only thing available. Table 1 lists several of the more common information integrity functions; a complete list would be much longer. All of these functions are mentioned in one or more of the chapters that make up this book. Some of them—such as authentication, digital signatures and shared capability—even have full chapters devoted to them.

TABLE 1 A PARTIAL LIST OF COMMON INFORMATION INTEGRITY FUNCTIONS

- Identification
- Authorization
- License and/or certification
- Signature
- Witnessing (notarization)
- Concurrence
- Liability
- Receipts
- Certification of origination and/or receipt
- Endorsement
- Access (egress)
- Validation
- Time of occurrence
- Authenticity—software and/or files
- Vote
- Ownership
- Registration
- Approval/disapproval
- Privacy (secrecy)

For example, there are many applications that need or even require a digital signature for digital information that would serve all the purposes now served by a handwritten signature to a document. There is no single technical means of solution to these problems, and, as a matter of fact, it remains an open question as to whether some of them even have feasible or legally acceptable solutions. There is a common element, however, to the solution to many of them, and that is cryptography or more precisely,

crypto-like transformations on the information whose integrity is to be insured. These are the technical means that make it possible for one or more parties who know a private piece (or pieces) of information to carry out an operation on the information which (probably) cannot be duplicated by someone not "in the know." The advantage or knowledge gained by being able to do this varies from application to application. In some cases it may be as simple as being granted access or entry to an automated teller machine (ATM) or to a remote computer or data bank; in others it may be the ability to conceal information or to recover hidden (encrypted) information, or it may be as complex as being able to "prove" to impartial third parties the culpability of a treaty signatory who has violated the terms of a treaty.

Simply put, information integrity is about how to prevent cheating, or failing that, to detect cheating in information-based systems wherein the information itself has no meaningful physical existence. Because there are so many different objectives for cheating where information is concerned, the subject of information integrity, and hence for the application of cryptographic principles, is consequently very broad. For example, the cheater may wish to impersonate some other participant in the system, or to eavesdrop on communications between other participants, or to intercept and modify information being communicated between other users of the system. The cheater may be an insider who either wishes to disavow communications that he actually originated or to claim to have received messages that were not sent. He may wish to enlarge his license to gain access to information that he has some level of authorized access for, or to subvert the system to alter (without authorization) the access license of others. The point is that since information can be enormously valuable or critical *so can its misuse.* Consequently, information integrity is concerned with devising means for either preventing or detecting all forms of cheating that depend on tampering with the information in information-based systems, where the means depend only on the information itself for their realization as distinguished from other noninformation-dependent means such as documentary records, physical security, etc.

Unless the reader has wrestled with real-world problems of protecting critical information from would-be cheaters, he is probably unaware of the gamut of reasons for cheating in information-based systems. Table 2 lists some of the more obvious reasons for cheating, each of which has arisen in one or more real-world situations. Not all these reasons have cryptographic solutions, but many do, and of these, most are discussed in one or more of the chapters in this book.

As mentioned earlier, the solution to problems of this type depends on the availability of operations (or transformations) on the information that is feasible for one or more participants in an information-based protocol to carry out because they know some private piece(s) of additional information, but which are (probably) impossible to do without knowing the private information. We will adopt this viewpoint to introduce the papers that make up *Contemporary Cryptology.*

In classical cryptography (secret key cryptography in the terminology used by Massey in his chapter of this book, "Contemporary Cryptology: An Introduction," or single-key cryptography in the terminology used by Brickell, Diffie, Moore, Odlyzko, and Simmons) there is only a single piece of private and necessarily secret information—the key—known to and used by the originator to encrypt information into a cipher and also known to and used by the intended recipient to decrypt the cipher. It is this operation of encryption and/or decryption that is assumed to (probably) be impossible to carry out without a knowledge of the secret key.

TABLE 2 REASONS FOR CHEATING

1. Gain unauthorized access to information, i.e., violate secrecy or privacy.
2. Impersonate another user either to shift responsibility, i.e., liability, or else to use his license for the purpose of:
   a. originating fraudulent information,
   b. modifying legitimate information,
   c. utilizing fraudulent identity to gain unauthorized access,
   d. fraudulently authorizing transactions or endorsing them.
3. Disavow responsibility or liability for information the cheater did originate.
4. Claim to have received from some other user information that the cheater created, i.e., fraudulent attribution of responsibility or liability.
5. Claim to have sent to a receiver (at a specified time) information that was not sent (or was sent at a different time).
6. Either disavow receipt of information that was in fact received, or claim a false time of receipt.
7. Enlarge his legitimate license (for access, origination, distribution, etc.).
8. Modify (without authority to do so) the license of others (fraudulently enroll others, restrict or enlarge existing licenses, etc.).
9. Conceal the presence of some information (a covert communication) in other information (the overt communication).
10. Insert himself into a communications link between other users as an active (undetected) relay point.
11. Learn who accesses which information (sources, files, etc.) and when the accesses are made (even if the information itself remains concealed), i.e., a generalization of traffic analysis from communications channels to data bases, software, etc.
12. Impeach an information integrity protocol by revealing information the cheater is supposed to (by the terms of the protocol) keep secret.
13. Pervert the function of software, typically by adding a convert function.
14. Cause others to violate a protocol by means of introducing incorrect information.
15. Undermine confidence in a protocol by causing apparent failures in the system.
16. Prevent communication among other users, in particular surreptitious interference to cause authentic communications to be rejected as unauthentic.

In public key cryptography, there are two pieces of information, at least one of which is computationally infeasible to recover from a knowledge of the other. One is the private piece of information (key) used by the originator to encrypt the information whose integrity is to be secured and the other is the private information (key) used by a recipient to decrypt the resulting ciphers. Depending on the application, both of these pieces of information need not be kept secret.

If it is computationally infeasible to recover the decryption key from the encryption key, then the encryption key need not be kept secret in order to insure the secrecy of the encrypted information using it. It must, however, be protected against substitution and/or modification, otherwise the transmitter could be deceived into encrypting information using a bogus encryption key for which the matching decryption key is known to an opponent (cheater). The decryption key must, of course, be kept secret and be physically secured against substitution and/or modification to insure the secrecy of the information concealed in the ciphers. This is the *secrecy channel.*

Conversely, if it is computationally infeasible to recover the encryption key from the decryption key, then the decryption key need not be kept secret. In this case, if a cipher, when decrypted, contains authenticating information (previously agreed on by the authorized transmitter or originator of the information and the intended recipients), then it was in all probability generated by the purported originator. This is the *authen-*

tication channel. The separation of these two functions by virtue of the separation of the two pieces of information needed to carry out the two complementary operations of encryption and decryption is the essential concept involved in public key cryptography, whose genesis is recounted by its inventor, Whitfield Diffie, in the chapter, "The First Ten Years of Public Key Cryptography."

One might at first think that this is the end of the process—that is, that having separated encryption and decryption and having put a computationally infeasible-to-overcome barrier between the pieces of information needed to carry them out, that nothing more is possible. To see this is not the case, one needs only to examine the list of reasons for cheating tabulated in Table 2. For example, if the party that is supposed to physically protect and keep secret the encryption key for an authentication channel, either deliberately or inadvertently allows it to be compromised (an example of deception #12 in Table 2), it then becomes impossible for an arbiter to establish who originated a cipher, even though the cipher contains the expected authenticating information. This example also illustrates the essential difference between actual signatures and digital signatures but more importantly it illustrates the first step in a natural "taxonomy of trust" in information integrity schemes described in detail in the chapter by Simmons, "A Survey of Information Authentication."

For commercial and private applications, probably the most important single information integrity function is a means to create digital signatures. As pointed out earlier, digital signatures differ in a critical respect from handwritten signatures because the author of a handwritten signature cannot transfer the ability to utter his signature to another party—no matter how great the desire to do so—while all that needs to be done to transfer the ability to utter the digital signature is to share the private piece of information used to generate it. Signature protocols can be devised to deal with this problem, reducing the likelihood of an attempted deception either being successful or else going undetected. Mitchell, Piper, and Wild provide a comprehensive treatment of the technical aspects of this topic in their chapter, "Digital Signatures." Because the applications for signatures (handwritten and digital) have to do with liability, concurrence, ownership, records, etc., all of which have legal implications, there is an evolving area of law concerned with the legal status and acceptability of digital signatures. A deliberate decision was made to limit the discussion here to the technical questions associated with creating digital signatures; however the reader should be aware that there are equally important, nontechnical issues.

In single-key cryptography, the transmitter and receiver have no choice but to trust each other unconditionally since either is capable of doing anything the other can. In the case of two-key cryptography only one specified participant (which can be either the transmitter or the receiver) must be assumed to be unconditionally trustworthy. The other participant is unable to carry out (some) actions that the other can, which means that the participant does not have to be trusted to not impersonate the other party insofar as those actions are concerned because he is not capable of doing so. But there are many applications in which no participant is *a priori* unconditionally trustworthy. It may, however, be reasonable to assume that some (unknown) elements in the system are trustworthy. Applications of this sort are discussed in the chapter "How to Insure That Data Acquired to Verify Treaty Compliance Are Trustworthy" by Simmons. As shown in that chapter, in order to prevent a unilateral action by one of the participants making it impossible to logically arbitrate disputes between mutually deceitful and distrusting parties, it becomes necessary for the operations on the information to depend on three

or more separate (but related) private pieces of information, all of which are necessary to correctly carry out the operations. The underlying idea is simple: to separate functional capability by separating the additional information needed to carry out the operations on the information, and then to give these separate pieces of information privately to the various participants in the protocol; in some cases to enable them to work cooperatively to carry out an operation, and in other cases to individually verify the authenticity of operations carried out by other participants. The logical extension of this notion of requiring the participation of three parties in order to carry out operations on information is to require the concurrence of specified—but arbitrary—subsets of the participants in order to do so. These concepts are discussed in detail in the chapter "An Introduction to Shared Secret and/or Shared Control Schemes and Their Application" by Simmons.

Cryptographic systems are commonly classified into block and stream ciphers—a rather artificial classification based on the size of the objects to which the cryptographic transformation is applied. If the objects are single symbols (normally an alphabetic or numeric character) the system is called a stream cipher, while if the object is made up of several symbols, the system is said to be a block cipher. In the second case, blocks could be considered to be symbols from a larger symbol alphabet, however the distinction between stream and block ciphers is a useful device when considering such problems as error propagation, synchronization, and especially of the achievable communication data rates and delays.

The search for, and often the catastrophic consequences of a failure to find, secure cryptoalgorithms for military and diplomatic applications, although conducted in great secrecy at the time, is well known up to a period shortly after World War II [2]. The development of algorithms for public use, however, has been carried out in full public view. The origins and subsequent development of what is certainly the best known, and arguably the most widely used, single-key cryptoalgorithm in history, the Data Encryption Standard (DES), are recounted by two of the principals in its adoption as a federal standard: Branstad and Smid, in their chapter "The Data Encryption Standard: Past and Future." An unusual aspect of this algorithm is that from its inception every detail of the DES operation has been public knowledge, an attribute common to almost all of the algorithms that have been the subject matter of the recent activity in cryptology.

Stream ciphers are of great practical importance, especially in applications where high data rates are required (secure video for example), and in which minimal communication delay is important. A comprehensive treatment of this subject is given by Rueppel in "Stream Ciphers." It should be pointed out that most fielded single-key secure communications technology is based on stream ciphers.

Diffie, in his chapter, "The First Ten Years of Public Key Cryptography," describes in detail the several attempts to devise secure two-key cryptoalgorithms and the gradual evolution of a variety of protocols based on them. A comprehensive treatment of this cornerstone of contemporary cryptology is given by Nechvatal in the chapter "Public Key Cryptography." Brickell and Odlyzko describe the efforts to disprove (or prove) the security of these schemes. Their chapter, "Cryptanalysis: A Survey of Recent Results," is the first compilation and cohesive presentation of the exciting sequence of cryptographic proposals and cryptanalytic breaks that have characterized public cryptology in the past decade, by two of the main contributors to those cryptanalytic successes. Rather than being discouraged by the cryptanalytic successes described there, one

should be encouraged by the emergence of algorithms and protocols whose security can be shown to be as "good" as some hard mathematical problem is difficult to solve, which is a new development in the science of cryptology. It is only the intense scrutiny and combined efforts of an active public research community that has brought this about. The bottom line is that after a decade and a half of effort, there are available acceptably secure single-key and two-key cryptoalgorithms in a variety of VLSI implementations whose operation is well understood and widely known. van Oorschot, in his chapter, "A Comparison of Practical Public Key Cryptosystems Based on Integer Factorization and Discrete Logarithms" gives a very thorough comparison of the relative merits of the principle contenders for two-key cryptoalgorithms—both from the algorithmic standpoint and from the efficiency of their best VLSI implementations to date. These comparisons (of apples and oranges to be sure) should be invaluable to a system designer faced with a choice among several algorithms, an even larger number of implementations, and of competing security, speed and protocol requirements.

As Massey points out in his chapter, "Contemporary Cryptography: An Introduction," even after suitable crypto-like operations have been devised, there still remain substantial cryptographic problems to be solved. How do the participants get the private pieces of information they need to perform their functions in the protocol and how can they be guaranteed of the integrity of what they receive? In an oversimplified form, this is the key distribution problem that was one of the stimuli for the discovery of public key cryptography (see the description by Diffie of the reasoning process that led to this discovery). The underlying problem, though, is broader than single-key distribution and is concerned with the entire question of how a participant in an information-based protocol can trust his part of the protocol and hence the soundness of the protocol itself, even though he cannot trust any of the other participants or the communications channel (data bank, software, etc.) from which the information is acquired. In its simplest form, this may reduce to how a user can be confident that his personal identification number (PIN) cannot be learned by someone at a financial institution and used to (undetectably) impersonate the user, or it may be as complex as how a participant can trust a nondeterministic, interactive, protocol between himself and a collection of other participants in which individual responses are complex functions of all of the prior responses, some of which are random, and in which the user must assume the other participants will collude to deceive or defraud him.

Even if one has a secure crypto-like operation or algorithm, and a trustworthy (that is, secure) means of distributing the private pieces of information to the participants, there is yet another way in which an information-based system can fail. These are protocol failures, discussed in the pioneering paper (and reprinted as a chapter in this book) "Protocol Failures in Cryptosystems" by Moore. Obviously, if the way private information is distributed in a protocol allows a compromise of information that should be kept secret, the cryptoalgorithm is broken, or if a collection of insiders can pool their private pieces of information to recover information that is supposed to be kept secret from them, then ordinary cryptanalysis may be possible. These sorts of failures, although potentially devastating to the integrity of a protocol, are not surprising, nor is their prevention particularly interesting. The cases of interest are those in which the intended function of the overall system or protocol can be defeated, even though the underlying cryptoalgorithm remains secure against cryptanalysis—that is, the system failure does not come about as a result of breaking the cryptoalgorithm. In a

sense, protocol failures are a result of cryptanalysis at the system level instead of at the algorithm level. As Moore makes clear through several examples, this type of failure occurs by exploiting information in unexpected ways. It is important, therefore, for understanding how cheating can occur in information-based systems, and hence, for understanding how to prevent cheating, to realize that information can be passed from one part of a protocol to another by a variety of channels other than the intended overt one.

One of the reasons the popular press has been so attracted by developments in contemporary cryptology is that many of the problems appear to be impossible to solve—making their solutions seem paradoxical. For example, problems such as how to make a single cipher mean different things to different people or how to conceal information in a cipher so that even someone who knows the cryptographic key used to produce the cipher will be unable to detect the presence of the concealed information have been solved. Other examples of seemingly impossible problems that also have been solved are how to authenticate a message even though nothing about the message can be kept secret from the very persons who wish to create fraudulent messages that would be accepted as authentic, or how to communicate securely despite the fact that none of the parties to the communication can be trusted. Perhaps the most paradoxical result of all is how one participant can prove to another that he knows a particular piece of information without revealing the information itself, and indeed without revealing anything about it that would aid someone else in pretending to know it. These protocols, which have formed the basis for a number of schemes for proof of identity, are introduced and discussed by Feigenbaum in her chapter "Overview of Interactive Proof Systems and Zero-Knowledge." Even after the concept is explained, the results still seem paradoxical. Interactive proof systems and zero-knowledge protocols are prototypes illustrating the impact of theoretical computer science on contemporary cryptology.

Nonspecialists are surrounded by transparent instances of information integrity schemes of the sort described here. They regularly identify themselves to ATMs, share access control to their safety deposit boxes with the institution, rely on the integrity of credit card numbers containing a low-level of security self-authenticating capability, etc. Less transparent examples are code-controlled scramblers on cable and/or satellite TV broadcasts, security for telephones (ranging from simple—and not very secure—analog schemes to the STU III NSA certified secure telephone units) etc. Almost everyone has daily contact with some of these information-integrity schemes; however, there is a new area of information technology that promises to eventually replace the ubiquitous plastic credit cards: smart cards. Smart cards that draw on several information-integrity technologies (cryptography, proof of identity, authentication, etc.) are described in the chapter, "Smart Card: A Standardized Security Device Dedicated to Public Cryptology" by three of the prime movers in their development: Guillou, Quisquater, and Ugon. This application will put a sophisticated information-integrity device in the wallet or purse of practically every person in the industrialized world, and will therefore probably be the most extensive application ever made of cryptographic schemes.

Finally, we note that our initial motivation in putting this book together and our concluding observation are the same; namely, that given the social, commercial, and personal importance of being able to protect information against all forms of information-based cheating, and given the apparent essential dependence of solutions to this class of problems on crypto-like transformations, it is desirable that computer scientists, communications engineers, systems designers, and others who may need to provide for the integrity of information and, of course, the ultimate end users who must

depend on the integrity of information, be acquainted with the essential concepts and principles of cryptography. The authors and the editor wish to thank the IEEE PRESS for their support in the publication of *Contemporary Cryptology* to satisfy this need.

References

[1] G. J. Simmons, "Cryptology," in *Encyclopedia Britannica,* 16th Edition. Chicago, IL: Encyclopedia Britannica Inc., pp. 913–924B, 1986.

[2] D. Kahn, *The Codebreakers.* New York: Macmillan, 1967 (abridged edition, New York: New American Library, 1974).

Contents

Contemporary Cryptology
An Introduction

JAMES L. MASSEY
Institute for Signal and Information Processing
Swiss Federal Institute of Technology
Zürich, Switzerland