CONTEMPORARY CRYPTOLOGY

The Science of Information Integrity

EDITED BY GUSTAVUS J. SIMMONS

IEEE PRESS

Previous Page Home Next page Next Chapter

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

- Privacy (Secrecy)
- Authentication
- Signatures (Digital)
- Identification
- Authorization
- License and/or Certification
- Witnessing (Notarization)
- Concurrence
- Liability
- Receipts
- Certification of Origination and/or Receipt
- Endorsement
- Access (Egress)
- Validation
- Time of Occurrence
- Vote
- Ownership
- Registration
- Approval/Disapproval
- Money

IEEE PRESS
445 Hoes Lane, PO Box 1331
Piscataway, NJ 088551331

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

ISBN B-87942-277-7
IEEE Order Number: PCO271-7

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 O-87942-277-7
1. Computer security. 2. Telecommunication systems-Security
measures. 3. Cryptography. 1. Simmons, Gustavus J.
QA76.9.A25C6678 1992
005.8’2-dc20
91-19684
CIP

Contents

Contemporary Cryptology: A Foreword vii
G. J. Simmons

Contemporary Cryptology: An Introduction 1
James L. Massey

Preliminaries . Secret key cryptography . Public key cryptography . Cryptographic protocols . . References

SECTION 1 CRYPTOGRAPHY 41

Chapter 1 The Data Encryption Standard: Past and Future 43
Miles E. Smid and Dennis K. Branstad

The birth of the DES . . . The DES controversy . . . Acceptance by government and commercial sectors . . Applications . New algorithms . . DES: The next decade . Conclusions . . . References

Chapter 2 **Stream Ciphers** 65
Rainer A. Raeppel

Introduction . Information-theoretic approach . . System-theoretic approach

. . *Complexity-theoretic approach . . . Randomized stream ciphers*
. *References*

Chapter 3 The First Ten Years of Public Key Cryptology l35
Whitfield Diffie

Initial discoveries . . . Exponential key exchange . Trapdoor knapsacks . The Rivest-Shamir-Adleman system . . . The McEliece coding scheme . The fall of the knapsacks . . . Early responses to public key cryptosystems . . . Application and implementation . . . Multiplying, factoring, and finding primes . . . Directions in public key cryptography research . . . Where is public key cryptography going? . . , References

Chapter 4 Public Key Cryptography 177
James Nechvatal

Introduction . . . Cryptosystems and cryptanalysis . . Key management . . . Digital signatures and hash functions . . . Examples of public key systems and hash functions . . . Implementations of public key cryptography . . . A sample proposal for a LAN implementation . . . Mathematical and computational aspects . An introduction to zero-knowledge . . Alternatives to the DifSie-Hellman model . . . Appendices , . . References

Chapter 5 A Comparison of Practical Public Key Cryptosystems Based on Integer Factorization and Discrete Logarithms 289
Paul C. van Oorschot

Introduction . . . Discrete logarithms in fields of characteristic 2 . . . Integer factorization . Comparing El Gamal in GF (2”) versus RSA . . . Recent work regarding elliptic curve cryptosystems . . . Concluding remarks . . . References

SECTION 2 AUTHENTICATION 323

Chapter 6 Digital Signatures 325
C. J. Mitchell, F. Piper, and P. Wild

Introduction . . . Fundamental concepts . . . Techniques for digital signatures . . . Techniques for hashing . . . Applications for digital signatures . . References

Chapter 7 A Survey of Information Authentication 379
G. J. Simmons

Introduction . . . The threat(s) . . . A natural classification of authentication schemes . . . How insecure can unconditionally secure authentication be? . . The practice of authentication . . . Conclusions . . References

SECTION 3 PROTOCOLS 421

Chapter 8 Overview of Interactive Proof Systems and Zero-Knowledge 423
J. Feigenbaum

Introduction . . . Definitions . . . Examples . . . Known results . . . Related notions . . Open problems . . . References , , . Appendix

Chapter 9 An Introduction to Shared Secret and/or Shared Control Schemes and Their Application 441
G. J. Simmons

Introduction . . . The general model(s) . . . Constructing concurrence schemes . . . The geometry of shared secret schemes . . . Setting up shared secret schemes . . Key distribution via shared secret schemes . . . Conclusions . . . References . . . Bibliography

SECTION 4 CRYPTANALYSIS 499

Chapter 10 Cryptanalysis: A Survey of Recent Results 501
E. F. Brickell and A. M. Odlyzko

Introduction . . . Knapsack cryptosystems . . . Generalized knapsack cryptosyterns. . . The Ong-Schnorr-Shamir (OSS) signature scheme . . . The Okamoto-Shiraishi signature scheme . . . Additional broken two-key systems . . . The RSA cryptosystem . . . Discrete exponentiation , . . The McEliece cryptosystem . , . Congruential generators . . . DES . . . Fast data encipherment algorithm . . . Additional comments . . . References

Chapter 11 Protocol Failures in Cryptosystems 541
J. H. Moore

Introduction . . . The notary protocol . . . The common modulus protocol . . The small exponent protocol failure . . . The low entropy protocol failure . . . A single key protocol failure . . . Summary and analysis . . References

SECTION 5 APPLICATIONS 559

Chapter 12 The Smart Card: A Standardized Security Device Dedicated to Public Cryptology 561
Louis Claude Guillou, Michel Ugon, and Jean-Jacques Quisquater

Introduction . . . Comprehensive approach Standardization . . . Technology Security Evolution of card authentication . . . Conclusions . . Appendix . . . Glossary References

Chapter 13 How to Insure That Data Acquired to Verify Treaty Compliance Are Trustworthy **615**
G. J. Simmons

Introduction . . . *Verification of a comprehensive test ban treaty . Verification without secrecy. . . Verification with arbitration . . . Verification in the presence of deceit . . . Concluding remarks*

Index **631**

Editor's Biography **640**

Contemporary Cryptology

A Foreword

Cryptology (from the Greek **kryptós**, “hidden,” and logos, “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 [l]. 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,

Previous Page Home Next page Next Chapter

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.

Previous Page Home Next page Next Chapter

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

Previous Page Home Next page Next Chapter

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

[l] 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).

Contemporary Cryptology Contemporary 1
PRELIMINARIES... 1
Introduction... 1
Cryptologic Nomenclature and Assumptions 2
The Need for Cryptology 2
Secret and Open Cryptologic Research................ 2
Epochs in Cryptology 3
Plan of This Chapter.. 7
SECRET KEY CRYPTOGRAPHY.................. 4
Model and Notation 4
Figure 1 Model of a secret key cryptosystem. 4
Theoretical and Practical Security 9
Perfect Secrecy 10
Key Requirements for Perfect Secrecy 11
Breaking an Imperfect Cipher 12
Authenticity and Deception 14
Figure 2 Modifications to Fig. 1 for consideration.. 15
Practical Security... 20
Diffusion and Confusion 20
The Data Encryption Standard 21
Stream Ciphers.. 22
Figure 3 A •typicalŽlinear-feedback 23
Provably Secure Ciphers? 24

PUBLIC KEY CRYPTOGRAPHY 25
One-Way Functions... 25
Public Key Distribution.................................... 26
The Rivest-Shamir-Adleman Public Key 27
Some Remarks on Public Key Cryptography .. 31

CRYPTOGRAPHIC PROTOCOLS 32
What Is a Protocol? 32
A Key-Distribution Protocol............................. 33
Shamir•s Three-Pass Protocol......................... 33
Conclusion 35

REFERENCES 36

Contemporary Cryptology

An Introduction

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

1. Preliminaries
2. Secret Key Cryptography
3. Public Key Cryptography
4. Cryptographic Protocols

An appraisal is given of the current status, both technical and nontechnical, of cryptologic research. The principal concepts of both secret-key and public key cryptography are described. Shannon's theory of secrecy and Simmons's theory of authenticity are reviewed for the insight that they give into practical cryptographic systems. Public key concepts are illustrated through consideration of the Diffie–Hellman public key-distribution system and the Rivest–Shamir–Adleman public key cryptosystem. The subtleties of cryptographic protocols are shown through consideration of some specific such protocols.

1 PRELIMINARIES

1.1 Introduction

That cryptology is a "hot" research area hardly needs saying. The exploits of cryptographic researchers are reported today not only in an increasing number of scholarly journals and popular scientific magazines, but also in the public press. One hears of conflicts between cryptologic researchers and government security agencies, insinuations of built-in "trapdoors" in commonly used ciphers, claims about new ciphers that would take millions of years to break and counterclaims that no cipher is secure—all the stuff of high drama. To ferret out the truth in such controversies, one needs a basic understanding of cryptology, of its goals and methods, and of its capabilities and limitations. The aim of this chapter is to provide a brief, self-contained introduction to cryptology that may help the reader to reach such a basic understanding of the subject, and that may give him or her additional insight into the more specialized papers on cryptology that form the rest of this book.

The present chapter is an updated, expanded, and slightly revised version of our earlier paper [45], large sections of which appear virtually unchanged herein. The reader who is familiar with this earlier paper may wish to concentrate his or her attention on the new material that appears in this one. Reference numbers [45] and onward denote references added to the earlier report and their appearance will flag such a reader's attention to the substantially new segments of this chapter.

Only scant attention will be given in this chapter to the long and rich history of cryptology. For an excellent short history, the reader is referred to that given in a splendid earlier survey of cryptology [1] or that in an unusually penetrating encyclopedia article [2]. But Kahn's voluminous history, *The Codebreakers* [3], is indispensable to anyone who wishes to dig deeply into cryptologic history. The abridged paperback edition [4] of Kahn's book can be especially recommended as it packs as much suspense as the best spy fiction has to offer, but will also satisfy the historical curiosity of most readers.

1.2 Cryptologic Nomenclature and Assumptions

The word *cryptology* stems from Greek roots meaning "hidden" and "word," and is the umbrella term used to describe the entire field of secret communications. For instance, the 8-year-old scientific society formed by researchers in this field is appropriately called the International Association for Cryptologic Research.

Cryptology splits rather cleanly into two subdivisions: cryptography and cryptanalysis. The cryptographer seeks to find methods to ensure the secrecy and/or authenticity of messages. The cryptanalyst seeks to undo the former's work by breaking a cipher or by forging coded signals that will be accepted as authentic. The original message on which the cryptographer plies his art is called the plaintext message, or simply the *plaintext;* the product of his labors is called the ciphertext message, or just the *ciphertext* or, most often, the *cryptogram*. The cryptographer always employs a *secret key* to control the enciphering process. Often (but not always) the secret key is delivered by some secure means (e.g., in an attaché case handcuffed to the wrists of a courier) to the person (or machine) to whom he expects later to send a cryptogram formed using that key.

The almost universal assumption of cryptography is that the enemy cryptanalyst has full access to the cryptogram. Almost as universally, the cryptographer adopts the precept, first enunciated by the Dutchman A. Kerckhoffs (1835–1903), that the security of the cipher must reside entirely in the secret key. Equivalently, *Kerckhoffs' assumption* is that the entire mechanism of encipherment, except for the value of the secret key, is known to the enemy cryptanalyst. If the cryptographer makes only these two assumptions, then he is designing the system for security against a *ciphertext-only attack* by the enemy cryptanalyst. If the cryptographer further assumes that the enemy cryptanalyst will have acquired ("by hook or by crook") some plaintext-cryptogram pairs formed with the actual secret key, then he is designing against a *known-plaintext attack*. The cryptographer may even wish to assume that the enemy cryptanalyst can submit any plaintext message of his own and receive in return the correct cryptogram for the actual secret key (a *chosen-plaintext attack*), or to assume that the enemy cryptanalyst can submit purported "cryptosystems" and receive in return the unintelligible garble to which they (usually) decrypt under the actual key (a *chosen-ciphertext attack*), or to assume both of these possibilities (a *chosen-text attack*). Most cipher systems in use today are intended by their designers to be secure against at least a chosen-plaintext attack, even if it is hoped that the enemy cryptanalyst will never have the opportunity to mount more than a ciphertext-only attack.

1.3 The Need for Cryptology

Cryptography has been used for millenia to safeguard military and diplomatic communications. Indeed, the obvious need for cryptography in the government sector led to the rather general acceptance, until quite recently, of cryptography as a prerogative of government. Most governments today exercise some control of cryptographic apparatus if not of cryptographic research. The United States, for instance, applies the same export/import controls to cryptographic devices as to military weapons. But the dawning of the Information Age revealed an urgent need for cryptography in the private sector. Today vast amounts of sensitive information such as health and legal records, financial transactions, credit ratings, and the like are routinely exchanged between computers via public communication facilities. Society turns to the cryptographer for help in ensuring the privacy and authenticity of such sensitive information.

While the need for cryptography in both the government and private sectors is generally accepted, the need for cryptanalysis is less well acknowledged. ''Gentlemen do not read each other's mail,'' was the response of U.S. Secretary of State H. L. Stimson in 1929 on learning that the U.S. State Department's ''Black Chamber'' was routinely breaking the coded diplomatic cables of many countries. Stimson forthwith abolished the Black Chamber, although as secretary of war in 1940 he relented in his distaste of cryptanalysis enough to condone the breaking of Japanese ciphers [4, p. 178]. In today's less innocent world, cryptanalysis is generally regarded as a proper and prudent activity in the government sector, but as akin to keyhole-peeping or industrial espionage in the private sector. However, even in the private sector, cryptanalysis can play a valuable and ethical role. The ''friendly cryptanalyst'' can expose the unsuspected weaknesses of ciphers so that they can be taken out of service or their designs remedied. A paradigm is Shamir's recent breaking of the Merkle–Hellman trapdoor-knapsack public key cryptosystem [5]. By publishing his ingenious cryptanalysis [6] of this clever and very practical cipher, Shamir forestalled its likely adoption in practice with subsequent exposure to the attacks of cryptanalysts seeking rewards more tangible than scientific recognition. Shamir's reward was the 1986 IEEE W. R. G. Baker Award.

In the preceding discussion, we abided by the long-accepted attribution of the dogmatic pronouncement, ''Gentlemen do not read each other's mail,'' to H. L. Stimson in 1929. Kruh [46] has recently given a convincing historical argument suggesting that these famous words may in fact have been uttered by Stimson first in 1946, rather than 1929, during his interviews with McGeorge Bundy, who was then preparing Stimson's authorized biography [47]. Kruh [46, p. 80] concludes, ''It thus seems highly likely that Stimson's 1946 remark accurately described his motivation for closing the Cipher Bureau in 1929. But whether he also said it then remains unknown.''

1.4 Secret and Open Cryptologic Research

If one regards cryptology as the prerogative of government, one accepts that most cryptologic research is conducted behind closed doors. Without doubt, the number of workers engaged today in such secret research in cryptology far exceeds that of those engaged in open research in cryptology. For only about 15 years has there in fact been widespread open research in cryptology. There have been, and will continue to be, conflicts between these two research communities. Open research is a common quest for knowledge that depends for its vitality on the open exchange of ideas via conference presentations and publications in scholarly journals. But can a government agency, charged with the responsibility of breaking the ciphers of other nations, countenance publication of a cipher that it could not break? Can a researcher in good conscience publish such a cipher that might undermine the effectiveness of his own government's code-breakers? One might argue that publication of a provably secure cipher would force all governments to behave like Stimson's ''gentlemen,'' but one must be aware that open research in cryptology is fraught with political and ethical considerations of a severity much greater than in most scientific fields. The wonder is not that some conflicts have occurred between government agencies and open researchers in cryptology, but rather that these conflicts (at least those of which we are aware) have been so few and so mild.

One can even argue that the greatest threat to the present vigorous open cryptologic research activity in the United States stems not from the intransigence of government but rather from its largesse. A recent U.S. government policy will require governmental agencies to rely on cryptographic devices at whose heart are tamper-proof modules incorporating secret algorithms devised by the National Security Agency (NSA) and loaded with master keys distributed by NSA [7]. Moreover, NSA will make these modules available to certified manufacturers for use in private-sector cryptography, and will presumably also supply the master keys for these applications. If, as appears likely, these systems find widespread acceptance in the American private sector, it will weaken the practical incentive for further basic open research in cryptography in the United States. The main practical application for such research will be restricted to international systems where the NSA technology will not be available.

1.5 Epochs in Cryptology

The entire period from antiquity until 1949 can justly be regarded as the *era of prescientific cryptology;* which is not to say that the cryptologic history of these times is devoid of interest today, but rather that cryptology was then plied almost exclusively as an art rather than as a science. Julius Caesar wrote to Cicero and his other friends in Rome more than 2000 years ago, employing a cipher in which each letter in the plaintext was replaced by the third (cyclically) later letter in the Latin alphabet [4, p. 77]. Thus, the plaintext CAESAR would yield the ciphertext FDHVDU. Today, we would express Caesar's cipher as

$$y = x \oplus z \tag{1}$$

where x is the plaintext letter ($A = 0, B = 1, \ldots, Z = 25$), z is the secret key (which Julius Caesar always chose as 3—Caesar Augustus chose 4), y is the ciphertext letter, and $\oplus$ here denotes addition modulo 26 (so that $23 \oplus 3 = 0$, $23 \oplus 4 = 1$, etc.). There is no historical evidence to suggest that Brutus broke Caesar's cipher, but a schoolchild today, who knew a little Latin and who had read the elementary cryptanalysis described in Edgar Allen Poe's masterful short story, ''The Gold Bug,'' would have no difficulty succeeding in a ciphertext-only attack on a few sentences of ciphertext. In fact, for the next almost two thousand years after Caesar, the cryptanalysts generally had a clear upper hand over the cryptographers. Then, in 1926, G. S. Vernam, an engineer with the American Telephone and Telegraph Company, published a remarkable cipher to be used with the binary Baudot code [8]. Vernam's cipher is similar to Caesar's in that it is described by Eq. (1), except that now x, y, and z take values in the binary alphabet $\{0, 1\}$ and $\oplus$ denotes addition modulo 2 ($0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 1 = 0$). The new idea advanced by Vernam was to *use the key only one time*, that is, to encipher each bit of plaintext with a new randomly chosen bit of key. This necessitates the secure transfer of as much secret key as one will later have plaintext to encipher, but it yields a truly unbreakable cipher as we shall see below. Vernam indeed believed that his cipher was unbreakable and was aware that it would not be so if the randomly chosen key bits were to be reused later, but he offered no proofs of these facts. Moreover, he cited in [8] field tests that had confirmed the unbreakability of his cipher, something no amount of field testing could in fact confirm. Our reason for calling the period up to 1949 the prescientific era of cryptology is that cryptologists then generally proceeded by intuition and "beliefs," which they could not buttress by proofs. It was not until the outbreak of World War II, for instance, that the English cryptologic community recognized that mathematicians might have a contribution to make to cryptology [9, p. 148] and enlisted among others, A. Turing, in their service.

The publication in 1949 by C. E. Shannon of the paper, "Communication Theory of Secrecy Systems" [10], ushered in the era of scientific secret key cryptology. Shannon, educated both as an electrical engineer and mathematician, provided a theory of secrecy systems almost as comprehensive as the theory of communications that he had published the year before [11]. Indeed, he built his 1949 paper on the foundation of the 1948 one, which had established the new discipline of information theory. Shannon not only proved the unbreakability of the random Vernam cipher, but also established sharp bounds on the required amount of secret key that must be transferred securely to the intended receiver when any perfect cipher is used.

For reasons that will become clear in the sequel, Shannon's 1949 work did not lead to the same explosion of research in cryptology that his 1948 report had triggered in information theory. The real explosion came with the publication in 1976 by W. Diffie and M. E. Hellman of their work, "New Directions in Cryptography" [12]. Diffie and Hellman showed for the first time that secret communications was possible without any transfer of a secret key between sender and receiver, thus establishing the turbulent *epoch of public key cryptography* that continues unabated today. R. C. Merkle, who had submitted his paper about the same time as Diffie and Hellman but to another journal, independently introduced some of the essential ideas of public key cryptography. Unfortunately, the long delay in publishing his work [13] has often deprived him of due scientific credit. A detailed first hand account by W. Diffie of this formative period for public cryptography is given in Chapter 3, "The First Ten Years of Public Key Cryptography," in this volume.

1.6 Plan of This Chapter

In the next section, we review briefly the theory of secret key cryptography, essentially following Shannon's original approach and making Shannon's important distinction between theoretic and practical security. We also indicate the directions of some contemporary research in secret key cryptography. Section 3 gives a short exposition of public key cryptography, together with a description of some of the most important public key

Previous Page Home Next page Next Chapter

systems thus far advanced. In Section 4 we touch on the delicate subject of cryptographic protocols, and show how cryptographic techniques can be used to accomplish nonstandard, but very useful, tasks.

2 SECRET KEY CRYPTOGRAPHY

2.1 Model and Notation

By a secret key cryptosystem, we mean a system that corresponds to the block diagram of Fig. 1. The essential feature of such a system is the "secure channel" by which the secret key, $Z = [Z_1, Z_2, \ldots, Z_K]$, after generation by the *key source*, is delivered to the intended receiver, protected from the prying eyes of the enemy cryptanalyst. To emphasize that the same secret key is used by both the encrypter and decrypter, secret key cryptosystems have also been called *one-key cryptosystems* and *symmetric cryptosystems*. The K digits of the key are letters in some finite alphabet that we will often choose to be the binary alphabet $\{0, 1\}$. The *message source* generates the plaintext, $X = [X_1, X_2, \ldots, X_M]$. The private random source (whose purpose will soon be evident) generates the *private randomizer*, $S = [S_1, S_2, \ldots, S_J]$, and the public random source (whose purpose will be seen later) generates the *public randomizer*, $R = [R_1, R_2, \ldots, R_T]$. The encrypter forms the cryptogram, $Y = [Y_1, Y_2, \ldots, Y_N]$, as a function of X, R, S, and Z. We write this encrypting transformation as

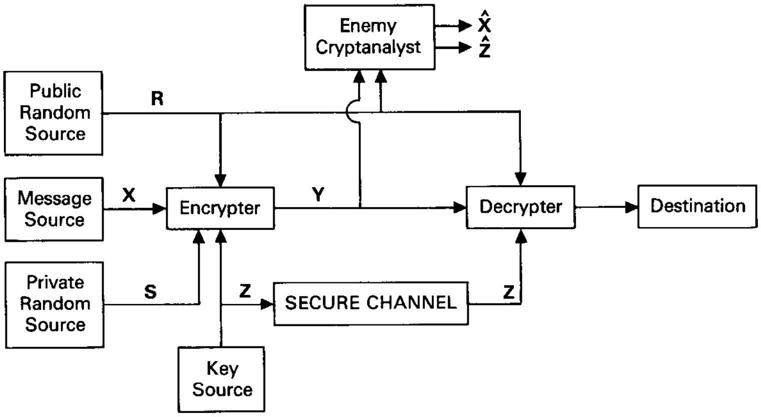

Figure 1 Model of a secret key cryptosystem.

$$Y = E_{ZRS}(X) \tag{2}$$

to underscore the fact that it is useful to think of the cryptogram Y as a function of the plaintext X with the particular function being specified by the values of the secret key Z and of the randomizing sequences R and S. As Fig. 1 implies, the decrypter must be able to invert this transformation without knowledge of the *private* randomizing sequence **S**. That is,

$$X = D_{ZR}(Y) \tag{3}$$

which expresses the fact that the plaintext X must be a function of the cryptogram Y where the particular function is determined only by the secret key Z and the public randomizer R. The enemy cryptanalyst observes the cryptogram Y and the public randomizer R but nothing else. The enemy cryptanalyst then forms an estimate $\hat{X}$ of the plaintext X and/or an estimate $\hat{Z}$ of the secret key Z. The enemy cryptanalyst, in accordance with Kerckhoff's precept, is assumed to know all details of the encrypter and decrypter, but of course to have no knowledge of X, S, and, in particular, of Z.

Our Fig. 1 differs from the "schematic of a general secrecy system" that appears as Fig. 1 in Shannon's 1949 paper [10] only in that we have included a private and a public randomizer in our model.

Private randomization is an old cryptographic trick. In English text the letter *e* appears much more frequently than any other letter. If English text is first converted into text in some larger alphabet by replacing *e* each time with a randomly chosen letter from the large "*e*-group" of letters in the larger alphabet, and similarly replacing other frequently chosen English letters with random choices of a letter from appropriately sized groups in the larger alphabet, one obtains a new text in which all letters of the larger alphabet have (approximately) the same frequency. Enciphering of this randomized text frustrates a single-letter frequency analysis by the enemy cryptanalyst. But, after deciphering the randomized text, the legitimate receiver can remove the randomization merely by replacing each letter in the *e*-group of the larger alphabet by the letter *e*, and so on—such a reader does not need to be told in advance which random substitutions would be made. Such randomized ciphers are known as "multiple-substitution ciphers" and also as "homophonic ciphers." The great mathematician, Gauss, deceived himself into believing that, by using homophonic substitution, he had devised an unbreakable cipher [2]; but, without question, private randomization is a useful cryptographic tool. We will see later that the newer cryptographic trick of using a public randomizer can be even more powerful in enhancing the security of a cryptographic system. For these reasons and because their inclusion scarcely complicates Shannon's theory of secrecy, we have included both types of randomizers in our Fig. 1.

It is important to recognize that X, Z, R, and S are *random quantities*. The statistics of the plaintext X are of course determined by the message source, but the statistics of the secret key Z and of the randomizing sequences R and S are under the control of the cryptographer. As Fig. 1 suggests, we shall always assume that the random quantities X, Z, R, and S are statistically independent.

2.2 Theoretical and Practical Security

Shannon considered two very different notions of security for cryptographic systems. He first considered the question of *theoretical security*, by which he meant, "How secure is a system against cryptanalysis when the enemy has unlimited time and manpower available for the analysis of intercepted cryptograms?" [10, p. 658]. Shannon's theory of theoretical security, which we shall review next, casts much light into cryptography, but leads to the pessimistic conclusion that the amount of secret key needed to build a theoretically secure cipher will be impractically large for most applications. Thus, Shannon also treated the question of *practical security*, by which he meant: Is the system secure against a cryptanalyst who has a certain limited amount of time and computational power available for the analysis of intercepted cryptograms? Public key systems, to be discussed in Section 3, are intended to provide practical security—they cannot provide theoretical security.

2.3 Perfect Secrecy

The first assumption in Shannon's theory of theoretical security is that the secret key will be used only one time, or equivalently that the M digits of the plaintext $\boldsymbol{X}$ form the total of messages that will be enciphered before the secret key $\boldsymbol{Z}$ and the randomizers $\boldsymbol{R}$ and S are changed. Because the enemy cryptanalyst observes only the cryptogram $\boldsymbol{Y}$ and the public randomizer $\boldsymbol{R}$, it is appropriate, following Shannon [10], to define *perfect secrecy* to mean that the plaintext $\boldsymbol{X}$ is statistically independent of the pair $\boldsymbol{Y}$ and $\boldsymbol{R}$, that is, that

$$P_{\boldsymbol{X}\,|\,\boldsymbol{YR}}(\boldsymbol{x}\,|\,\boldsymbol{y},\boldsymbol{r}) = P_{\boldsymbol{X}}(\boldsymbol{x})$$

holds for all $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{r}$. This is the same as saying that the enemy cryptanalyst can do no better estimating $\boldsymbol{X}$ with knowledge of $\boldsymbol{Y}$ and $\boldsymbol{R}$ than could be done in the absence of this knowledge, no matter how much time and computing power the enemy cryptanalyst has at his disposal. Having made the right mathematical formulation of the problem, it was then child's play for Shannon to show that perfect secrecy systems exist.

Consider the case of a nonrandomized cipher in which the plaintext, ciphertext, and key digits all takes values in the L-ary alphabet $\{0, 1, \ldots, L-1\}$, and in which the length K of the key and length N of the cryptogram coincide with the length M of the plaintext, that is, $K = N = M$. Suppose that the key is chosen to be *completely random*, that is, $P(\boldsymbol{Z} = \boldsymbol{z}) = L^{-M}$ for all L^M possible values $\boldsymbol{z}$ of the secret key, and that the enciphering transformation is

$$Y_i = X_i \oplus Z_i, \qquad i = 1, 2, \ldots, M \tag{4}$$

where $\oplus$ denotes addition mod L. Because for each possible choice x_i and y_i of X_i and Y_i, respectively, there is a unique z_i such that $Z_i = z_i$ satisfies Eq. (4), it follows that $P(\boldsymbol{Y} = \boldsymbol{y}\,|\,\boldsymbol{X} = \boldsymbol{x}) = L^{-M}$ for every possible particular $\boldsymbol{y}$ and $\boldsymbol{x}$, no matter what the statistics of $\boldsymbol{X}$ may be. Thus $\boldsymbol{X}$ and $\boldsymbol{Y}$ are statistically independent, and hence this *modulo-L Vernam system* (to use Shannon's terminology) provides perfect secrecy. The modulo-L Vernam system is better known under the name, the *one-time pad*, from its use shortly before, during, and after World War II by spies of several nationalities who were given a pad of paper containing the randomly chosen secret key and told that it could be used for only one encipherment. There appears to have been a general belief in cryptographic circles that this cipher was unbreakable, but Shannon seems to have been the first to publish a proof of this theoretical unbreakability.

It is worth noting here that the one-time pad offers perfect secrecy no matter what the statistics of the plaintext $\boldsymbol{X}$ may be. In fact, we will show shortly that it also uses the least possible amount of secret key for any cipher that provides perfect secrecy independent of the statistics of the plaintext—this is a most desirable attribute; one would not usually wish the security of the cipher system to depend on the statistical nature of the message source. But the fact that the one-time pad requires one digit of secret key for each digit of plaintext makes it impractical in all but the few cryptographic applications, such as encrypting the Moscow–Washington hotline, where the need for secrecy is paramount and the amount of plaintext is quite limited.

We have learned recently from a reliable source that the Washington–Moscow hotline is no longer encrypted with a one-time pad, but that in its stead a conventional secret-key cipher that requires much less key is used. This change is apparently the result of increased confidence within the closed cryptographic community in the security of the secret key ciphers at their disposal.

2.4 Key Requirements for Perfect Secrecy

To go further in the study of theoretical security, we need to make use of some properties of "uncertainty" (or "entropy"), the fundamental quantity in Shannon's information theory [11]. *Uncertainty* is always defined as the mathematical expectation of the negative logarithm of a corresponding probability distribution. For instance, $H(X \mid Y)$ (which should be read as "the uncertainty about X given knowledge of Y") is the expectation of the negative logarithm of $P_{X \mid Y}(X \mid Y)$, that is,

$$H(X \mid Y) = \sum_{xy \,\varepsilon\, \text{supp}\,(P_{XY})} P_{XY}(x,y)\,(-\log P_{X \mid Y}(x \mid y))$$

where supp (P_{XY}) denotes the set of all x, y such that $P_{XY}(x, y) \neq 0$. (The reason that in information theory one takes an expectation by summing only over the *support* of the joint probability distribution of the random variables involved is that this permits one to deal with the expectation of functions such as $-\log P_{X \mid Y}(x \mid y)$ that can take on the values $-\infty$ or $+\infty$.) Uncertainties obey intuitively pleasing rules, such as $H(X, Y) = H(X) + H(Y \mid X)$, which we will use in our discussion of theoretical secrecy without further justification—the reader is referred to [11] or to the introductory chapters of any standard textbook on information theory for proofs of the validity of these "obvious" manipulations of uncertainties.

Equations (2) and (3) can be written equivalently in terms of uncertainties as

$$H(Y \mid X,Z,R,S) = 0 \tag{5}$$

and

$$H(X \mid Y,R,Z) = 0 \tag{6}$$

respectively, because, for instance, $H(Y \mid X,Z,R,S)$ is zero if and only if X,Z,R, and S together uniquely determine Y. Shannon's definition of perfect secrecy can then be written as

$$H(X \mid Y,R) = H(X) \tag{7}$$

since this equality holds if and only if X is statistically independent of the pair Y and R.

For any secret key cryptosystem, one has

$$\begin{aligned} H(X \mid Y,R) &\leq H(X,Z \mid Y,R) \\ &= H(Z \mid Y,R) + H(X \mid Y,R,Z) \end{aligned}$$

$$= H(\boldsymbol{Z} \mid \boldsymbol{Y},\boldsymbol{R}) \\ \le H(\boldsymbol{Z}) \qquad (8)$$

where we have made use of Eq. (6) and of the fact that the removal of given knowledge can only increase uncertainty. If the system gives perfect secrecy, it follows from Eqs. (7) and (8) that

$$H(\boldsymbol{Z}) \ge H(\boldsymbol{X}) \qquad (9)$$

Inequality [Eq. (9)] is *Shannon's fundamental bound for perfect secrecy; the uncertainty of the secret key must be at least as great as the uncertainty of the plaintext that it is concealing*. If the K digits in the key are chosen from an alphabet of size L_z, then

$$H(\boldsymbol{Z}) \le \log (L_z^K) = K \log L_z \qquad (10)$$

with equality if and only if the key is completely random. Similarly,

$$H(\boldsymbol{X}) \le M \log L_x \qquad (11)$$

(where L_x is the size of the plaintext alphabet) with equality if and only if the plaintext is completely random. Thus, if $L_x = L_z$ (as in the one-time pad) and if the plaintext is completely random, Shannon's bound [Eq. (9)] for perfect secrecy yields, with the aid of Eq. (10) and of equality in Eq. (11),

$$K \ge M \qquad (12)$$

That is, the key must be at least as long as the plaintext, a lower bound that holds with equality for the one-time pad.

2.5 Breaking an Imperfect Cipher

Shannon also considered the question of when the enemy cryptanalyst would be able in theory to break an imperfect cipher. To this end, he introduced the *key equivocation function*

$$f(n) = H(\boldsymbol{Z} \mid Y_1, Y_2, \ldots, Y_n) \qquad (13)$$

which measures the uncertainty that the enemy cryptanalyst has about the key given that he has examined the first n digits of the cryptogram. Shannon then defined the *unicity distance u* as the smallest n such that $f(n) \approx 0$. Given u digits of the ciphertext and not before, there will be essentially only one value of the secret key consistent with $Y_1, Y_2, \ldots, Y_n$, so it is precisely at this point that the enemy cryptanalyst with unlimited time and computing power could deduce the secret key and thus break the cipher. Shannon showed for a certain well-defined "random cipher" that

$$u \approx \frac{H(\boldsymbol{Z})}{r \log L_y} \qquad (14)$$

where

$$r = 1 - \frac{H(\boldsymbol{X})}{N \log L_y} \qquad (15)$$

Previous Page Home Next page Next Chapter

is the *percentage redundancy* of the message information contained in the N digit cryptogram, whose letters are from an alphabet of size L_y. When $N = M$ and $L_x = L_y$ (as is true in most cryptosystems), r is just the percentage redundancy of the plaintext itself, which is about $\frac{3}{4}$ for typical English text. When $L_x = L_z$ and the key is chosen completely at random to maximize the unicity distance, Eq. (14) gives

$$u \approx \frac{K}{r} \tag{16}$$

Thus, a cryptosystem with $L_x = L_y = L_z$ used to encipher typical English text can be broken after only about $N = \frac{4}{3}K$ ciphertext digits are received. For instance, a secret key of 56 bits (8 American Standard Code for Information Interchange [ASCII] 7-bit symbols) can be found in principle from examination of only about 11 ASCII 7-bit symbols of ciphertext.

Although Shannon's derivation of Eq. (14) assumes a particular kind of "random" cipher, he remarked "that the random cipher analysis can be used to estimate equivocation characteristics and the unicity distance for the ordinary types of ciphers" [10, p. 698]. Wherever it has been possible to test this assertion of Shannon's, it has been found to be true. Shannon's approximation [Eq. (14)] is routinely used today to estimate the unicity distance of "ordinary" secret key ciphers.

The reader may well be worrying about the validity of Eqs. (14) and (16) when $r = 0$, as it would in the case when $N = M$, $L_x = L_y$, and the message source emitted completely random plaintext so that $H(X) = M \log L_x = N \log L_y$. The answer is somewhat surprising: The enemy cryptanalyst can never break the system ($u = \infty$ is indeed the correct unicity distance!), even if $K \ll M$ so that Eq. (12) tells us that the system does not give perfect secrecy. The resolution of this paradox is that perfect secrecy demands that Y provide no information at all about X, whereas breaking the system demands that Y determines X essentially uniquely, that is, that Y must provide the maximum possible information about X. If the secret key Z were also chosen completely at random in the cipher for the completely random message source described above, there would always be L_z^K different plaintext-key pairs consistent with any possible cryptogram y, and all would be equally likely alternatives to the hapless cryptanalyst. This suggests, as Shannon was quick to note, that *data compression is a useful cryptographic tool.* An ideal data compression algorithm transforms a message source into the completely random (or "nonredundant") source that we have just been considering. Unfortunately, no one has yet devised a data compression scheme for realistic sources that is both ideal and practical (nor is anyone ever likely to do so), but even a nonideal scheme can be used to decrease r significantly, and thus to increase the unicity distance u significantly. Experience had long ago taught cryptographers that redundancy removal was a useful trick. In the days when messages were hand-processed, cryptographers would often delete from the plaintext many letters and blanks that could be recognized as missing and be replaced by the legitimate receiver. THSISASIMPLFORMOFDATACOMPRESION.

Shannon's derivation of Eq. (14) assumed a cryptographic system without the two randomizers that we have included in our Fig. 1. When a private randomizer S is included in the system, then $H(X)$ in Eq. (15) must be replaced by the joint uncertainty $H(X,S)$ for Eq. (14) still to hold.This suggests that randomization can also be used to reduce the redundancy r in the cryptogram. This, too, old-time cryptographers had learned from experience. They frequently inserted extra symbols into the plaintext, often an X, to hide the real statistics of the message. THXISISAXNEXAMXPLE.

Homophonic substitution, which was discussed in Section 2.1, is also a method for using a private randomizer to reduce the redundancy r in the cryptogram. Günther [48] quite recently suggested an ingenious variant of homophonic substitution in which the substitutes for a single plaintext letter are binary strings of varying length. Günther showed that it is possible to make the redundancy of the ciphertext *exactly* zero while at the same time making only a modest expansion in the number of binary digits needed to represent the plaintext. Jendahl, Kuhn, and Massey [49] modified Günther's scheme to achieve the minimum possible expansion of the plaintext and showed that, on the average, less than 4 bits of a completely random binary private randomizer suffice to determine the homophonic string for replacing each plaintext letter (whether or not the plaintext alphabet is also binary). What keeps both of these schemes from achieving zero redundancy in practice (and hence from yielding unbreakable practical ciphers) is that both schemes require complete and exact knowledge of the plaintext statistics, something that is never available for real information sources. However, both schemes can make use of available partial knowledge of the plaintext statistics (such as knowledge of the statistics of single letters, pairs of letters, and triplets of letters) to reduce greatly the redundancy r of the cryptogram and hence to increase greatly the unicity distance of an "ordinary" cipher.

2.6 Authenticity and Deception

We have mentioned several times that cryptography seeks to ensure the secrecy and/or authenticity of messages. But it is in fact quite a recent realization that secrecy and authenticity are independent attributes. If one receives a cryptogram that decrypts under the actual secret key to a sensible message, cannot one be sure that this cryptogram was sent by one's friend who is the only other person privy to this secret key? The answer, as we shall see, in general is: No! The systematic study of authenticity is the work of G. J. Simmons [14], who has developed a theory of authenticity that in many respects is analogous to Shannon's theory of secrecy.

To treat the theoretical security of authenticity systems as formulated by Simmons, we must give the enemy cryptanalyst more freedom than he is allowed in the model of Fig. 1. Figure 2 shows the necessary modification to Fig. 1. The enemy cryptanalyst is now the one who originates the "fraudulent" cryptogram $\tilde{Y}$ that goes to the decrypter. The line from the decrypter to the destination is shown dotted in Fig. 2 to suggest that the decrypter might recognize $\tilde{Y}$ as fraudulent and thus not be deceived into passing a fraudulent plaintext $\tilde{X}$ to the destination.The authentic cryptogram Y is shown on a dotted input line to the enemy cryptanalyst in Fig. 2 to suggest that the latter may have to form his fraudulent cryptogram $\tilde{Y}$ without ever seeing the authentic cryptogram itself.

As did Shannon, Simmons assumes that the secret key Z will be used only one time, that is, to form only one authentic cryptogram Y. But Simmons recognized that even in this case, three quite different attacks need to be distinguished. First, the enemy may of necessity form a fraudulent cryptogram $\tilde{Y}$ without knowledge of the authentic cryptogram Y [the *impersonation attack*], indeed Y might not yet exist. The impersonation attack is said to succeed if the decrypter accepts $\tilde{Y}$ as a valid cryptogram—even if it should turn out later that $\tilde{Y}$ coincides with the valid cryptogram Y. The *probability of successful impersonation,* P_I, is defined as the enemy's probability of success when he

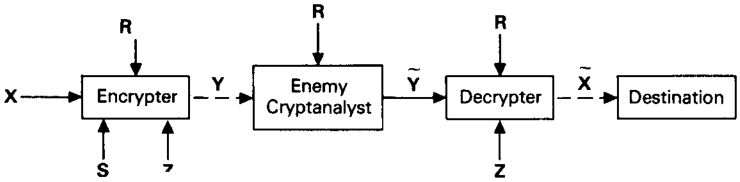

Figure 2 Modifications to Fig. 1 for consideration of authenticity attacks.

or she employs an optimum impersonation strategy. Second, the enemy cryptanalyst may be able to intercept the authentic cryptogram Y and replace it with his fraudulent cryptogram $\tilde{Y}$ where $\tilde{Y} \neq Y$ [the *substitution attack*]. The substitution attack succeeds if the decrypter accepts $\tilde{Y}$ as a valid cryptogram, and the *probability of successful substitution, P_S,* is defined as the probability of success when the enemy employs an optimum substitution strategy. And third, the enemy may be able to choose freely between an impersonation attack and a substitution attack [the *deception attack*]; the *probability of successful deception, P_d,* is then defined as the probability of success for an optimum deception strategy.

It may appear obvious that $P_d = \max(P_I, P_S)$. Simmons, however, used a game-theoretic authentication model, which was appropriate for the treaty-compliance-and-verification problems that he was considering and in which the cryptographer has the freedom to choose the key statistics to foil the type of attack that the enemy cryptanalyst may choose. In this case, one can only assert that $P_d \geq \max(P_I, P_S)$, since the best choice of key statistics for foiling a deception attack can differ from that for foiling an impersonation attack or for foiling a substitution attack. Our adoption of Kerckhoffs' assumption (see Section 2, above), however, forces us to assume that the key statistics are fixed once and for all by the cryptographer, independently of the attack used by the enemy cryptanalyst. In this case, which we assume hereafter, it is indeed true that $P_d = \max(P_I, P_S)$.

The theory of authenticity is in many ways more subtle than the corresponding theory of secrecy. In particular, it is not at all obvious how "perfect authenticity" should be defined. Let $\#\{Y\}$ denote the number of cryptograms y such that $P(Y = y) \neq 0$, and let $\#\{X\}$ and $\#\{Z\}$ be similarly defined as the number of plaintexts and keys, respectively, with nonzero probability. It follows from Eq. (3) that, for every z, there must be at least $\#\{X\}$ different cryptograms y such that $P(Y = y \mid Z = z) \neq 0$. Hence, if the enemy cryptanalyst in an impersonation attack selects Y completely at random from the $\#\{Y\}$ cryptograms with nonzero probability, his probability of success will be at least $\#\{X\}/\#\{Y\}$. Thus, P_I, the probability of success in an optimal impersonation attack satisfies

$$P_I \geq \#\{X\}/\#\{Y\} \tag{17}$$

This equation shows that good protection against an impersonation attack demands that $\#\{Y\}$ be much greater than $\#\{X\}$, and shows that *complete protection* (that is, $P_I = 0$) *is impossible.* We note further that Eq. (17) can hold with equality only when there are exactly $\#\{X\}$ valid cryptograms y for each key z, which means that a randomized cipher cannot achieve equality in Eq. (17).

Because complete protection against deception is impossible, the only recourse is to define "perfect authenticity" to mean as much protection against deception as is possible given the size of the space of valid cryptograms (even if this means that we must call a system "perfect" for which $\#\{Y\} = \#\{X\}$ and hence $P_d = 1$). This is what Simmons has done, but we must develop the theory a little further before introducing his precise definition of "perfect authenticity."

Let the *authentication function*, $\phi(y, z)$ be defined to be 1 if y is a valid cryptogram for the secret key z and to be 0, otherwise. Note that if $Z = z$ the decrypter will accept $\tilde{Y} = y$ as a valid cryptogram just when $\phi(y, z) = 1$. The probability that a particular y is a valid cryptogram can be written

$$P(y \text{ valid}) = \sum_z \phi(y,z) P_Z(z) \tag{18}$$

which is just the total probability of the keys z for which y is a valid cryptogram. The best impersonation attack is for the enemy cryptanalyst to choose $\tilde{Y} = y$ for that y that maximizes $P(y$ valid). Thus

$$P_I = \max_y P(y \text{ valid}) \tag{19}$$

In [14], Simmons derived the following fundamental lower bound on the probability of successful impersonation:

$$P_I \geq 2^{-I(Y;\, Z)} \tag{20}$$

which reveals the quite surprising fact that P_I can be made small *only if the cryptogram gives away much information about the secret key*—at least in principle, exploiting this information is another matter. One of the minor original contributions of our earlier paper [45] was a shortened proof of the bound [Eq. (20)] that allowed one to identify the necessary and sufficient conditions for equality. This simplification motivated Sgarro [50] to provide a still simpler proof of Eq. (2) based on properties of "informational divergence," Johannesson and Sgarro then observed that the bound [Eq. (20)] could be strengthened and, in their paper [51] thereon, included an even simpler proof of Eq. (20) that was suggested to them by Körner and is based on the "logsum inequality" [52, p. 48]. This led in turn to our finding yet a new proof of Eq. (20) that we now present.

It is immediately apparent from Eq. (19) that

$$P_I \geq \sum_y P_Y(y) P(y \text{ valid}) \tag{21}$$

with equality if and only if $P(y$ valid) is constant for all y. Substituting Eq. (18) into Eq. (21) gives

$$P_I \geq \sum_{yz} P_Y(y) P_Z(z) \phi(y,z) \tag{22}$$

But the pair y and z is in supp P_{YZ} precisely when $\phi(y,z) = 1$ and $P_Z(z) \neq 0$. Thus, this last inequality can be written equivalently in terms of an expectation as

$$P_I \geq E\left[\frac{P_Y(y)P_Z(z)}{P_{YZ}(y,z)}\right]$$

as follows from the discussion of expectations in Section 2.4, above. This inequality is of course equivalent to

$$\log P_I \geq \log E\left[\frac{P_Y(y)P_Z(z)}{P_{YZ}(y,z)}\right] \tag{23}$$

Because the logarithm is a strictly concave function, Jensen's well-known inequality [15, pp. 151–152] can be applied to give

$$\log E\left[\frac{P_Y(y)P_Z(z)}{P_{YZ}(y,z)}\right] \geq E\left[\log \frac{P_Y(y)P_Z(z)}{P_{YZ}(y,\ z)}\right] \tag{24}$$

with equality if and only if $(P_Y(y)P_Z(z))/P_{YZ}(y,z)$ is constant for all pairs y and z in supp P_{YZ}. The final step in the derivation of Eq. (20) is to note that

$$E\left[\log \frac{P_Y(y)P_Z(z)}{P_{YZ}(y,z)}\right] = H(Y,\ Z) - H(Y) - H(Z) = -\ I(Y;\ Z) \tag{25}$$

Combining Eqs. (23)–(25) gives

$$\log P_I \geq -\ I(Y;\ Z) \tag{26}$$

which is equivalent to Eq. (20). The necessary and sufficient conditions for equality in Eq. (20) are seen to be that both (i) $P(y$ valid) is constant for all y (or, equivalently, that *every impersonation strategy is optimum*) and (ii) that $(P_Y(y)P_Z(z))/P_{YZ}(y,z)$ is constant for all pairs y and z in supp P_{YZ}.

Johannesson and Sgarro [51] obtained a first strengthening of Simmons's bound [Eq. (20)] by noting that although P_I does not depend on the statistics of the plaintext X (as can be seen from Eqs. (18) and (19)), the mutual information $I(Y;\ Z)$ generally does depend on P_X. Thus

$$P_I \geq 2^{-\text{inf1}\ I(Y;\ Z)} \tag{27}$$

where inf1 here denotes the *infimum* (or "minimum") of $I(Y;\ Z)$ over all choices of P_X that leaves the authentication function $\phi(y,z)$ unchanged. They further strengthened this bound by noting that nothing in the derivation of Eq. (20) demands that the plaintext X and the key Z be statistically independent (although they always are in our model and in practice) and hence that

$$P_I \geq 2^{-\text{inf2}\ I(Y;\ Z)} \tag{28}$$

where inf2 denotes the infimum of $I(Y;\ Z)$ over all choices of *conditional* probability distributions for the plaintext X given the key Z.

Because for our Kerckhoffian assumption we have that

$$P_d = \max(P_I,\ P_S) \tag{29}$$

it follows that Eq. (20) gives also *Simmons's lower bound on the probability of successful deception*, namely,

$$P_d \geq 2^{-I(Y;\ Z)} \tag{30}$$

where conditions (i) and (ii) above are necessary, but no longer sufficient, conditions for equality.

Simmons [14] has defined *perfect authenticity* to mean that equality holds in Eq. (30). Even with perfect authenticity, however, it must be remembered that the probability of deception P_d will be small only when $I(\mathbf{Y}; \mathbf{Z})$ is large; that is, only when the cryptogram provides the enemy cryptanalyst with much information about the key! The information that $\mathbf{Y}$ gives about $\mathbf{Z}$ is a measure of how much of the secret key is used to provide authenticity.

[It might seem more appropriate to define "perfect authenticity" to mean that equality holds when the stronger bounds inf1 $I(\mathbf{Y}; \mathbf{Z})$ or inf2 $I(\mathbf{Y}; \mathbf{Z})$ are used on the right of Eq. (30). However, it seems to us better to abide by Simmons's use of $I(\mathbf{Y}; \mathbf{Z})$ and then to consider the case when inf1 $I(\mathbf{Y}; \mathbf{Z})$ or inf2 $I(\mathbf{Y}; \mathbf{Z})$ is less than $I(\mathbf{Y}; \mathbf{Z})$ as indicating that the authenticity system is "wasting" part of the information $I(\mathbf{Y}; \mathbf{Z})$ that the cryptogram $\mathbf{Y}$ betrays about the key $\mathbf{Z}$ and thus does not deserve the appellation "perfect."]

The theory of the theoretical security of authenticity systems is less well developed than is that of secrecy systems. In particular, it is not known in general under what conditions systems offering perfect authenticity exist, although constructions of particular such systems have been given. Thus, we will content ourselves here with giving a series of simple examples that illuminate the main ideas of authentication theory and show the relation between authentication and secrecy.

In the following examples, the plaintext is always a single binary digit X, the cryptogram $\mathbf{Y} = [Y_1, Y_2]$ a binary sequence of length 2, the key $\mathbf{Z} = [Z_1, \ldots, Z_K]$ is a completely random binary sequence so that $P(\mathbf{Z} = \mathbf{z}) = 2^{-K}$ for all $\mathbf{z}$, and all logarithms are taken to the base 2 so that $H(\mathbf{Z}) = K$ bits.

Example 1. Consider the encipherment scheme with a key of length $K = 1$ described by the following table.

z \ x	0	1
0	00	10
1	01	11

The meaning is that, for instance, $Y = [1, 0]$ when $X = 1$ and $Z = 0$. The enciphering rule is simply $\mathbf{Y} = [X, Z]$, that is, the key is appended as a "signature" to the plaintext to form the cryptogram. Thus, this system provides *no secrecy* at all. Moreover, $H(Z \mid \mathbf{Y}) = 0$ so that $I(\mathbf{Y}; \mathbf{Z}) = 1$ bit, and the bound [Eq. (28)] becomes $P_I \geq \frac{1}{2}$. But if $P_X(0) = \frac{1}{2}$, then $P(\mathbf{y} \text{ valid}) = \frac{1}{2}$ for all $\mathbf{y}$ so that in fact $P_I = \frac{1}{2}$, which is as small as possible. But upon observing $\mathbf{Y} = \mathbf{y}$, the enemy cryptanalyst always knows the other valid cryptogram so that he can always succeed in a substitution attack. Hence $P_S = 1 = P_d > 2^{-I(\mathbf{Y}; \mathbf{Z})} = \frac{1}{2}$, that is, the authenticity is not perfect.

Example 2. Consider the randomized encipherment system in which the private randomizer S is a binary random variable with $P(S = 0) = \frac{1}{2}$.

z	s \ x	0	1
0	0	00	10
0	1	01	11
1	0	00	11
1	1	01	10

Previous Page Home Next page Next Chapter

Note that $Y_1 = X$ so that again there is *no secrecy.* Given $\boldsymbol{Y} = \boldsymbol{y}$ for any $\boldsymbol{y}$, the two possible values of $\boldsymbol{Z}$ are equally likely so that $H(\boldsymbol{Z} \mid \boldsymbol{Y}) = 1$, and thus $I(\boldsymbol{Y}; \boldsymbol{Z}) = 0$. It follows then from Eq. (28) that this system must have $P_I = 1 = P_d = 2^{-I(Y;Z)}$ and thus trivially provides perfect authenticity. But on observing, say, $\boldsymbol{Y} = [0, 0]$, the enemy cryptanalyst is faced with two equally likely alternatives, [1, 0] and [1, 1], for the other valid cryptogram, only one of which will be accepted by the receiver, who knows $\boldsymbol{Z}$, as authentic. Thus $P_S = \frac{1}{2}$. This example shows that a randomized cipher can satisfy Eq. (30) with equality, and also that $-I(\boldsymbol{Y}; \boldsymbol{Z})$ is *not* in general a lower bound on $\log P_S$.

Examples 1 and 2 show that *the substitution attack can be stronger than the impersonation attack*, and *vice versa.*

Example 3. Consider the same system as in Example 2 except that z and s are now the two digits z_1 and z_2, respectively, of the secret key, and hence both are known to the legitimate receiver. There is still *no secrecy* because $Y_1 = X$. Given $\boldsymbol{Y} = \boldsymbol{y}$ for any $\boldsymbol{y}$, there are still two equally likely possibilities for $\boldsymbol{Z}$ so that $H(\boldsymbol{Z}|\boldsymbol{Y}) = 1$ and hence $I(\boldsymbol{Y}; \boldsymbol{Z}) = 1$ bit. But $P(\boldsymbol{y}$ valid$) = \frac{1}{2}$ for all four cryptograms $\boldsymbol{y}$ and thus $P_I = \frac{1}{2}$. Moreover, given that he observes $\boldsymbol{Y} = \boldsymbol{y}$, the enemy cryptanalyst is faced with two equally likely choices for the other valid cryptogram so that $P_S = \frac{1}{2}$. Thus $P_d = \frac{1}{2} = 2^{-I(Y;Z)}$ and hence this system offers (nontrivial) *perfect authenticity,* no matter what the statistics of the plaintext X may be.

Example 4. Consider the following encipherment system.

	x		
z_1	z_2	0	1
0	0	00	11
0	1	01	10
1	0	10	01
1	1	11	00

Because $P(\boldsymbol{Y} = \boldsymbol{y} \mid X = x) = \frac{1}{4}$ for all x and $\boldsymbol{y}$, the system provides *perfect secrecy.* By the now familiar arguments, $I(\boldsymbol{Y}; \boldsymbol{Z}) = 1$ bit and $P_I = \frac{1}{2}$, the corresponding best possible protection against impersonation when $H(X) = 1$, that is, when $P(X = 0) \; = \frac{1}{2}$. But, on observing $\boldsymbol{Y} = \boldsymbol{y}$, the enemy can always succeed in a substitution attack by choosing $\boldsymbol{Y}$ to be the complement of $\boldsymbol{y}$. Thus $P_S = 1 = P_d$ and hence this system provides *no protection against deception* by substitution.

Example 5. Consider the following encipherment system.

	x		
z_1	z_2	0	1
0	0	00	10
0	1	01	00
1	0	11	01
1	1	10	11

This cipher provides *perfect secrecy* and has $I(\boldsymbol{Y}; \boldsymbol{Z}) = 2 - H(\mathrm{X})$. Moreover, $P(\boldsymbol{y}$ valid$) = \frac{1}{2}$ for all $\boldsymbol{y}$ so that $P_I = \frac{1}{2}$. Upon observing that $\boldsymbol{Y} = \boldsymbol{y}$, say $\boldsymbol{y} = [0, 0]$, the enemy cryptanalyst is faced with the two alternatives [1,0] and [0,1] for the other valid cryptogram with the probabilities $P(X = 0)$ and $P(X = 1)$, respectively. Thus, $P_S \geq \frac{1}{2}$ with equality if and only if $P(X = 0) = \frac{1}{2}$. It follows that $P_d = P_S \geq 2^{-I(Y;Z)} = \frac{1}{2}$ with equality if and only if $P(X = 0) = \frac{1}{2}$. Thus, if $P(X = 0) = \frac{1}{2}$, this cipher also provides *perfect authenticity.*

Examples 3, 4, and 5 illustrate the fact that *secrecy and authenticity are independent attributes of a cryptographic system*—a lesson that is too often forgotten in practice.

2.7 Practical Security

In Section 2.5, we noted the possibility for a cipher system with a limited key [i.e., with $K \ll H(X)$] to have an infinite unicity distance and hence to be theoretically "unbreakable." Shannon called such ciphers *ideal,* but noted that their design poses virtually insurmountable practical problems [10, p. 700]. Most practical ciphers must depend for their security not on the theoretical impossibility of their being broken, but on the practical difficulty of such breaking. Indeed, Shannon postulated that every cipher has a *work characteristic* $W(n)$ which can be defined as the average amount of work (measured in some convenient units such as hours of computing time on a CRAY 2) required to find the key when given n digits of the ciphertext. Shannon was thinking here of a ciphertext-only attack, but a similar definition can be made for any form of cryptanalytic attack. The quantity of greatest interest is the limit of $W(n)$ as n approaches infinity, which we shall denote by $W(\infty)$ and which can be considered the average work needed to "break the cipher." Implicit in the definition of $W(n)$ is that the *best possible cryptanalytic algorithm* is employed to break the cipher. Thus to compute or underbound $W(n)$ for a given cipher, we are faced with the extremely difficult task of finding the best possible way to break that cipher, or at least of proving lower bounds on the work required in the best possible attack. There is no practical cipher known today (at least to researchers outside the secret research community) for which even an interesting lower bound on $W(\infty)$ is known. Such practical ciphers are generally evaluated in terms of what one might call the *historical work characteristic,* $W_h(n)$, which can be defined as the average amount of work to find the key from n digits of ciphertext when one uses the best of *known attacks* on the cipher. When one reads about a "cipher that requires millions of years to break," one can be sure that the writer is talking about $W_h(\infty)$. When calculated by a cryptographer who is fully acquainted with the techniques of cryptanalysis, $W_h(\infty)$ can be a trustworthy measure of the real security of the cipher, particularly if the cryptographer includes a judicious "margin of error" in his calculations. But there always lurks the danger that $W(\infty) \ll W_h(\infty)$, and hence that an enemy cryptanalyst might devise a new and totally unexpected attack that will, when it is ultimately revealed, greatly reduce $W_h(\infty)$—the history of cryptography is rife with examples!

2.8 Diffusion and Confusion

Shannon suggested two general principles, which he called diffusion and confusion [10, p. 708], to guide the design of practical ciphers. By *diffusion,* he meant the spreading out of the influence of a single plaintext digit over many ciphertext digits so as to hide the statistical structure of the plaintext. An extension of this idea is to spread the influence of a single key digit over many digits of ciphertext so as to frustrate a piecemeal attack on the key. By *confusion,* Shannon meant the use of enciphering transformations that complicate the determination of how the statistics of the ciphertext depend on the

Previous Page Home Next page Next Chapter

statistics of the plaintext. But a cipher should not only be difficult to break, it must also be easy to encipher and decipher when one knows the secret key. Thus, a very common approach to creating diffusion and confusion is to use a *product cipher,* that is, a cipher that can be implemented as a succession of simple ciphers, each of which adds its modest share to the overall large amount of diffusion and confusion.

Product ciphers most often employ both transposition ciphers and substitution ciphers as the component simple ciphers. A *transposition cipher* merely permutes the letters in the plaintext, the particular permutation being determined by the secret key. For instance, a transposition cipher acting on six-letter blocks of Latin letters might cause CAESAR to encipher to AESRAC. The single-letter statistics of the ciphertext are the same as for the plaintext, but the higher-order statistics of the plaintext are altered in a confusing way. A *substitution cipher* merely replaces each plaintext letter with another letter from the same alphabet, the particular substitution rule being determined by the secret key. The single-letter statistics of the ciphertext are the same as for the plaintext. The Caesar cipher discussed in Section 1.5 is a simple substitution cipher with only 26 possible values of the secret key. But if the substitution is made on a very large alphabet so that it is not likely that any plaintext letter will occur more than once in the lifetime of the secret key, then the statistics of the plaintext are of little use to the enemy cryptanalyst and a substitution cipher becomes quite attractive. To achieve this condition, the cryptographer can choose the ''single letters'' on which the substitution is applied to be groups of several letters from the original plaintext alphabet. For instance, a substitution upon pairs of Latin letters, in which CA was replaced by WK, ES by LB, and AR by UT, would result in CAESAR being enciphered to WKLBUT. If this ciphertext was then further enciphered by the above-considered transposition cipher, the resulting ciphertext would be KLBTUW. Such interleaving of simple transpositions and substitutions, when performed many times, can yield a very strong cipher, that is, one with very good diffusion and confusion.

2.9 The Data Encryption Standard

Perhaps the best example of a cipher designed in accordance with Shannon's diffusion and confusion principles is the Data Encryption Standard (DES). In the DES, the plaintext X, the cryptogram Y, and the key Z are binary sequences with lengths $M = 64$, $N = 64$, and $K = 56$, respectively. All 2^{64} possible values of X are, in general, allowed. Because $M = N = 64$, DES is in fact a substitution cipher, albeit on a very large alphabet of $2^{64} \approx 10^{19}$ ''letters''! In its so-called *electronic code book mode,* successive 64-bit ''blocks'' of plaintext are enciphered using the same secret key, but otherwise independently. Any cipher used in this manner is called a *block cipher.*

The DES is a product cipher that employs 16 ''rounds'' of successive encipherment, each round consisting of rather simple transpositions and substitutions of 4-bit groups. Only 48 key bits are used to control each round, but these are selected in a random-appearing way for successive rounds from the full 56-bit key. We shall not pursue further details of the DES here; a good short description of the DES algorithm appears in [1] and the complete description is readily available [16]. It suffices here to note that it appears hopeless to give a useful description of how a single plaintext bit (or a single key bit) affects the ciphertext (good diffusion!), or of how the statistics of the plaintext affect those of the ciphertext (good confusion!)

The DES algorithm was submitted by the IBM Corporation in 1974 in response to the second of two public invitations by the U.S. National Bureau of Standards (NBS) for designers to submit algorithms that might be used as a standard for data encryption by government and private entities. One design requirement was that the algorithm could be made public without compromising its security—a requirement that Kerckhoffs would have admired! The IBM design was a modification of the company's older Lucifer cipher that used a 128-bit key. The original design submitted by IBM permitted all $16 \times 48 = 768$ bits of key used in the 16 rounds to be selected independently. A U.S. Senate Select Committee ascertained in 1977 that the NSA was instrumental in reducing the DES secret key to 56 bits that are each used many times, although this had previously been denied by IBM and NBS [17]. NSA also classified the *design principles* that IBM had used to select the particular substitutions that are used within the DES algorithm. But the entire algorithm in full detail was published by NBS in 1977 as a U.S. Federal Information Processing Standard [16], to become effective in July of that year.

Almost from the beginning, the DES was embroiled in controversy. W. Diffie and M. E. Hellman, cryptologic researchers at Stanford University, led a chorus of skepticism over the security of the DES that focused on the smallness of the secret key. With $2^{56} \approx 10^{17}$ possible keys, a *brute-force attack* or ''exhaustive cryptanalysis'' (in which the cryptanalyst tries one key after another until the cryptogram deciphers to sensible plaintext) on the DES was beyond feasibility, but only barely so. Diffie and Hellman published the conceptual blueprint for a highly parallel special-purpose computer that, by their reckoning, would cost about 20 million dollars and would break DES cryptograms by essentially brute force in about 12 hours [18]; Hellman later proposed a variant machine, that, by his reckoning, would cost only four million dollars and, after a year of initial computation, would break 100 cryptograms in parallel each day [19]. Countercritics have attacked both of these proposals as wildly optimistic. But the hornet's nest of public adverse criticism of DES did lead the NBS to hold workshops of experts in 1976 and 1977 to ''answer the criticisms'' [17] and did give rise to the Senate hearing mentioned above. The general consensus of the workshops seems to have been that DES would be safe from a Diffie–Hellman-style attack for only about ten years, but that the 56-bit key provided no margin of safety [17]. Almost fifteen years have now passed, and the DES appears to have justified the faith of its defenders. Despite intensive scrutiny of the DES algorithm by cryptologic researchers, no one has yet publicly revealed any weakness of DES that could be exploited in an attack that would be significantly better than exhaustive cryptanalysis. The general consensus of cryptologic researchers today is that DES is an extremely good cipher with an unfortunately small key. But it should not be forgotten that the effective size of the secret key can be increased by using multiple DES encryptions with different keys, that is, by making a product cipher with DES used for the component ciphers. At least three encryptions should be used to foil the ''meet-in-the-middle attack'' proposed by Merkle and Hellman [20].

2.10 Stream Ciphers

In a block cipher, a plaintext block identical to a previous such block would give rise to the identical ciphertext block as well. This is avoided in the so-called *stream ciphers* in which the enciphering transformation on a plaintext ''unit'' changes from unit to unit.

For instance, in the *cipher-block chaining* (CBC) mode proposed for the DES algorithm [16], the current 64-bit plaintext block is added bit-by-bit modulo 2 to the previous 64 bit ciphertext block to produce the 64-bit block that is then enciphered with the DES algorithm to produce the current ciphertext block. CBC converts a block cipher into a stream cipher with the advantage that tampering with ciphertext blocks is more readily detected, that is, impersonation or substitution attacks become much more difficult. But cryptographers generally reserve the term *stream cipher* for use only in the case when the plaintext "units" are very small, say a single Latin letter or a single bit.

The most popular stream ciphers today are what can be called *binary additive stream ciphers.* In such a cipher, the K bit secret key $\mathbf{Z}$, is used only to control a *running-key generator* (RKG) that emits a binary sequence, Z'_1, Z'_2, . . . , Z'_N , called the *running key*, where in general $N \gg K$. The ciphertext digits are then formed from the binary plaintext digits by simple modulo 2 addition in the manner

$$Y_n = X_n \oplus Z'_n, \qquad n = 1, 2, \ldots N \tag{31}$$

Because modulo 2 addition and subtraction coincide, Eq. (31) implies

$$X_n = Y_n \oplus Z'_n, \qquad n = 1, 2, \ldots N \tag{32}$$

which shows that encryption and decryption can be performed by identical devices. A single plaintext bit affects only a single ciphertext bit, which is the worst possible diffusion; but each secret key bit can influence many ciphertext bits so the key diffusion can be good.

There is an obvious similarity between the binary additive stream cipher and a binary one-time pad. In fact, if $Z_n = Z'_n$ (that is, if the secret key is used as the running key), then the additive stream cipher is identical to the one-time pad. This similarity undoubtedly accounts in part for the widespread faith in additive stream ciphers that one encounters in many cryptographers and in many users of ciphers. But, of course, in practical stream ciphers, the ciphertext length N greatly exceeds the secret key length K. The best that one can then hope to do is to build an RKG whose output sequence cannot be distinguished by a resource-limited cryptanalyst from a completely random binary sequence. The trick is to build the RKG in such a way that, on observing Z'_1, Z'_2, . . . , Z'_n, the resource-limited cryptanalyst can do no better than to guess Z'_{n+1} at random. If this can be done, one has a cipher that is secure against even a chosen-plaintext attack (by which one would mean that the enemy cryptanalyst could freely select, say the first n bits of the plaintext sequence).

Stream ciphers have the advantage over block ciphers in that analytic measures of their quality are more easily formulated. For instance, stream cipher designers are greatly concerned with the *linear complexity* or "linear span" of the running-key sequence, which is defined as the length L of the shortest linear-feedback shift-register (LFSR) that could produce the sequence. Figure 3 shows a typical LFSR of length 6.

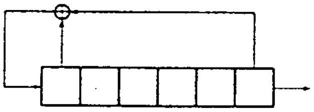

Figure 3 A "typical" linear-feedback shift-register.

$\oplus$ = modulo 2-adder.

Previous Page Home Next page Next Chapter

The reason for this concern is that there is a simple algorithm that would quickly find this shortest LFSR after examining only the first $2L$ bits of the running key [21]. Thus, large linear complexity of the running-key sequence is a necessary (but far from sufficient) condition for the practical security of an additive stream cipher. (An up-to-date treatment of linear complexity in connection with stream ciphers may be found in the book by Rueppel [44].) The RKG of an additive stream cipher is often built by the nonlinear combining of the output sequences of several LFSRs, as such combining can create a sequence with large linear complexity. There arises then the danger that individual LFSR sequences will be correlated with the running-key sequence so that the enemy cryptanalyst can attack the cipher piecemeal. Siegenthaler [22] has shown recently that the "correlation-immunity" of nonlinear combining functions can be precisely quantified and that the designer has to make an explicit tradeoff between correlation-immunity and linear complexity. There are many other known analytic approaches to stream cipher design. Taken together, they still leave one far from the point where one could say that the true work characteristic of a practical stream cipher is known, but they tend to give many cryptographers and users (perhaps misleadingly) greater trust in the historical work characteristics computed for stream ciphers than in those computed for block ciphers.

2.11 Provably Secure Ciphers?

When dealing with the practical security of ciphers, "It is difficult to define the pertinent ideas involved with sufficient precision to obtain results in the form of mathematical theorems," as Shannon said nearly 40 years ago [10, p. 702] in an eloquent understatement that needs no alteration today. It is an open question whether it is even possible to compute the true work characteristic $W(n)$ or its asymptotic value $W(\infty)$. A slender ray of hope lies in a totally impractical cipher proposed by this writer and I. Ingemarsson [23]. This cipher is a randomized stream cipher with a secret key of K bits. One can prove that $W(\infty) \approx 2^{K/2}$ where the unit of computation is a binary test, that is, a test with two outcomes. The "catch" is that the legitimate receiver must wait (during which time he does no testing or other computational work) until about 2^K bits have arrived before deciphering is begun. One can easily guarantee that the enemy cryptanalyst will need thousands of years to break the cipher, if one is willing to wait millions of years to read the plaintext! Such a cipher would be tolerable perhaps only to Rip van Winkle, the lazy and sleep-prone hero of Washington Irving's delightful short story, after whom both the story and the cipher have been named. Randomization, which was the feature that allowed the calculation of $W(\infty)$ for the impractical Rip van Winkle cipher, may turn out to be useful in developing a practical provably secure cipher, if in fact this can be done at all.

The previous words, which appeared in our earlier paper [45], have taken on a prophetic ring. At Eurocrypt'90, Maurer [53] presented a new cipher that exploits a very large public randomizer $\boldsymbol{R}$, that is provably secure, and that is at least arguably on the verge of being practical. Perhaps the most interesting facet of Maurer's work was his introduction of a new information-theoretic approach to cryptography that allows one to overcome the "bottleneck" of Shannon's inequality [Eq. (9)] for perfect secrecy. Maurer's trick was to introduce a *security event*, S, with the property that the cipher provides *perfect secrecy given that the event S occurs* [and even if $H(X) \gg H(Z)$]—but "all bets are off" when S does not occur! For his "strongly randomized" cipher, Mau-

Previous Page Home Next page Next Chapter

rer showed that the probability that S does *not* occur will be negligibly small unless the enemy cryptanalyst examines a substantial fraction of all the bits in the very large public randomizer $\boldsymbol{R}$. The legitimate sender and receiver need examine only the very small portion of the public randomizer that is specified by the short secret key $\boldsymbol{Z}$. The conclusion from Maurer's work is the (in retrospect obvious) fact that Shannon's bound [Eq. (9)] governs the needed key size only when one demands that his cipher provide *perfect secrecy with probability 1.*

3 PUBLIC KEY CRYPTOGRAPHY

3.1 One-Way Functions

That the publication of Shannon's 1949 paper [10] resulted in no discernible upsurge in open cryptologic research is due to several factors. First, the theory of theoretical security of secrecy systems that it provided was virtually complete in itself, and showed conclusively that theoretically secure secrecy systems demand the secure transfer of far more secret key than is generally practicable. Moreover, the insights that Shannon provided into the practical security of secrecy systems tended to reinforce accepted cryptographic approaches rather than to suggest new ones. But Shannon's observation that "The problem of good cipher design is essentially one of finding difficult problems, subject to certain other conditions. . . . We may construct our cipher in such a way that breaking it is equivalent to (or requires at some point in the process) the solution of some problems known to be laborious" [10, p. 704] took root in the fertile imaginations of the Stanford cryptologic researchers, W. Diffie and M. E. Hellman. The fruit was their 1976 paper, "New Directions in Cryptography," [12] that stunned the cryptologic world with the startling news that *practically secure secrecy systems can be built that require no secure transfer of any secret key whatsoever.*

The crucial contribution of the Diffie–Hellman paper lies in two unusually subtle definitions, that of a "one-way function," which was borrowed from work by R. M. Needham on secure computer login techniques, and that of a "trapdoor one-way function," which was totally new. A *one-way function* is defined as a function f such that for every x in the domain of f, $f(x)$ is easy to compute; but for virtually all y in the range of f, it is computationally infeasible to find an x such that $y = f(x)$. The first thing to note is that this is not a precise mathematical definition. What do "easy," "virtually all" (which we have substituted for Diffie and Hellman's "almost all," as the latter can have a precise mathematical meaning that was not intended in the definition), and "computationally infeasible" mean precisely? Yet the definition is sufficiently precise that one has no doubt as to what Diffie and Hellman essentially meant by a one-way function, and one has the feeling that it could be made completely precise in a particular context. It is less clear how such a function could be of use cryptographically—to build a cipher that not even the legitimate receiver could decipher seems the obvious (and worthless) application! A *trapdoor one-way* function is defined as a family of invertible functions f_z, indexed by z, such that, given z, it is easy to find algorithms E_z and D_z that easily compute $f_z(x)$ and $f_z^{-1}(y)$ for all x and y in the domain and range, respectively, of f_z; but for virtually all z and for virtually all y in the range of f_z, it is computationally infeasible to compute $f_z^{-1}(y)$ even when one knows

E_z. Again, this is only a semimathematical definition, but this time the cryptologic utility is nakedly apparent.

3.2 Public Key Distribution

As a likely candidate for a one-way function, Diffie and Hellman [12] suggested the *discrete exponential function*

$$f(x) = \alpha^x \pmod p \tag{33}$$

where x is an integer between 1 and $p - 1$ inclusive, where, as indicated, the arithmetic is done modulo p, a very large prime number, and where $\alpha(1 \le \alpha < p)$ is an integer such that $\alpha, \alpha^2, \ldots, \alpha^{p-1}$ are, in some order, equal to $1, 2, \ldots, p - 1$. For example, with $p = 7$, one could take $\alpha = 3$ since $\alpha = 3$, $\alpha^2 = 2$, $\alpha^3 = 6$, $\alpha^4 = 4$, $\alpha^5 = 5$, and $\alpha^6 = 1$. (In algebraic terminology, such an α is called a *primitive element* of the finite field $GF(p)$, and such α's are known always to exist.) If $y = \alpha^x$, then it is natural to write

$$x = \log_\alpha (y) \tag{34}$$

so that inverting $f(x)$ is the problem of calculating *discrete logarithms*. Even for very large p, say $p \approx 2^{1000}$, it is quite easy to calculate $f(x)$ by the trick of square-and-multiply. For instance, to compute $\alpha^{53} = \alpha^{32+16+4+1}$, one would first form α^2, $\alpha^4 = (\alpha^2)^2$, $\alpha^8 = (\alpha^4)^2$, $\alpha^{16} = (\alpha^8)^2$, and $\alpha^{32} = (\alpha^{16})^2$, which requires five multiplications. Then one would multiply α^{32}, α^{16}, α^4, and α together, which takes three more multiplications for a total of eight multiplications (mod p). Even with $p \approx 2^{1000}$, calculation of $f(x)$ for any integer x, $1 \le x < p$, would take less than 2000 multiplications (mod p).

If the discrete exponential function is indeed one-way, then for virtually all integers y, $1 \le y < p$, it must be computationally infeasible to compute $\log_x y$. It was soon realized by Hellman and Pohlig that it was not enough that p be large, $p - 1$ must also have a large prime factor (ideally, $p - 1$ would be twice another prime) if the discrete logarithm is indeed to be hard to compute [24]. With this proviso, the best of known algorithms for computing the discrete logarithm require very roughly (see [28] for precise details) $\sqrt{p}$ multiplies (mod p), compared to only about $2 \log_2 p$ multiplies for discrete exponentiation. If the discrete logarithm is truly this hard to compute, then the discrete exponential with the proviso on $p - 1$ is indeed a one-way function. But as of this writing *there is no proof that the discrete exponential, or any other function for that matter, is truly one-way.*

Diffie and Hellman suggested an astoundingly simple way in which the discrete exponential could be used to create secret keys between pairs of users in a network using only public messages. All users are presumed to know α and p. Each user, say user i, randomly selects an integer X_i between 1 and $p - 1$ that is kept as his *private secret*. The user then computes

$$Y_i = \alpha^{X_i} \pmod p \tag{35}$$

Rather than keeping Y_i secret, the user places Y_i in a *certified public directory* accessible to all users. If users i and j later wish to communicate secretly, user i fetches Y_j from the directory, then uses the private secret X_i to form

$$Z_{ij} = (Y_j)^{X_i} = (\alpha^{X_j})^{X_i} = \alpha^{X_i X_j} \pmod p \tag{36}$$

In a similar manner, user j forms Z_{ji}. But $Z_{ij} = Z_{ji}$ so that users i and j can now use Z_{ij} as the secret key in a conventional cryptosystem. If an enemy could solve the discrete logarithm problem, he could take Y_i and Y_j from the directory, solve for $X_i = log_\alpha Y_i$, and then form Z_{ij} in the same manner as did user i—there seems to be no other way for an enemy to find Z_{ij} (but there is no proof of this). The scheme just described is the Diffie–Hellman *public key-distribution system.* Although it is the oldest proposal for eliminating the transfer of secret keys in cryptography, it is still generally considered today to be one of the most secure and most practical public key schemes.

It should not be overlooked that the Diffie–Hellman public key-distribution scheme (and indeed every public key technique) *eliminates the need for a secure channel to pass along secrets, but does not eliminate the need for authentication.* The custodian of the public directory must be certain that it is indeed user i who puts the (nonsecret) Y_i into the directory, and user i must be certain that Y_i was actually sent to him by the custodian of the public directory. But it must not be forgotten that in secret key cryptography, (see Fig. 1) the receiver must not only be sure that the key Z was kept secret en route to him, but also that the key Z was actually sent by the legitimate sender. *Public key methods* remove one of these two problems; they *do not create a new authentication problem,* but rather make the old authentication problem more apparent.

3.3 The Rivest–Shamir–Adleman Public Key Cryptosystem

Having defined a trapdoor one-way function, it was an easy step for Diffie and Hellman to propose the structure of a *public key cryptosystem* for a network of many users. Each user, say user i, randomly chooses a value Z_i of the index and keeps Z_i as his *private secret.* The user next forms the algorithm E_{Z_i} which is then *published* in the certified public directory. Each user also forms the algorithm D_{Z_i} that is *kept secret* for each user's own use. If user j wishes to send a secret message X to user i, he fetches E_{Z_i} from the directory. User j then uses this algorithm to compute the cryptogram $Y = f_{Z_i}(X)$ that is then sent to user i. User i uses his private algorithm D_{Z_i} to compute $f_{Z_i}^{-1}(Y) = X$. If f_z is truly a trapdoor one-way function, this cryptosystem provides unassailable practical security.

When, for every index z, the domain and range of f_z coincide, Diffie and Hellman noted that a trapdoor one-way function can be used to create *digital signatures.* If user i wishes to send a *nonsecret* message X (to any or all users in the system) that he wishes to "sign" in a way that the recipient will recognize him unmistakably as the author, he merely uses his private algorithm to form $Y = f_{Z_i}^{-1}(X)$ and transmits Y. Every user can fetch the public algorithm E_{Z_i} and then compute $f_{Z_i}(Y) = X$; but no one except user i could have known how to write an intelligible message X in the form $Y = f_{Z_i}^{-1}(X)$, since no one except user i can compute $f_{Z_i}^{-1}$. Of course, user i could also send a signed secret message to user j by encrypting Y in user j's public key E_{Z_j}, rather than sending Y in the clear (he might first need to break Y into smaller pieces if Y is "too large to fit" into the domain of f_{Z_j}).

It was not at all clear to Diffie and Hellman in 1976 whether trapdoor one-way functions existed, and they did not hazard a conjectured such function in their paper. It was left to R. L. Rivest, A. Shamir, and L. Adleman (RSA) of the Massachusetts Institute of Technology (MIT) to make the first proposal of a possible trapdoor one-way function in their remarkable 1978 work, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" [25]—it is interesting to note that authentication received higher billing than secrecy in their title. The RSA trapdoor one-way function is the essence of simplicity, but to describe it we need a few ideas from elementary number theory.

Let GCD(i, n) denote the greatest common divisor of the integers i and n (not both 0). For example, GCD(12, 18) = 6. The *Euler totient function* $\phi(n)$, where n is a positive integer, is defined as the number of positive integers i less than n such that GCD(i, n) = 1 (except that $\phi(1)$ is defined to be 1). For instance, $\phi(6) = 2$ since for $1 \leq i < 6$ only $i = 1$ and $i = 5$ give GCD(i, 6) = 1. One sees immediately that for a prime p, $\phi(p) = p - 1$; and just a little thought more shows that if p and q are distinct primes, then

$$\phi(pq) = (p - 1)(q - 1) \tag{37}$$

For instance, $\phi(6) = \phi(2 \times 3) = 1 \times 2 = 2$. A celebrated theorem of Euler (1707–1783) states that for any positive integers x and n with $x < n$

$$x^{\phi(n)} = 1 \pmod{n} \tag{38}$$

provided that GCD(x, n) = 1. For example

$$5^2 = 1 \pmod{6}$$

The last fact from number theory that we need is that if e and m satisfy $0 < e < m$ and GCD(m, e) = 1, then there is a unique d such that $0 < d < m$ and

$$de = 1 \pmod{m} \tag{39}$$

moreover d can be found in the process of using Euclid's "extended" algorithm for computing GCD(m, e), (see [26, p. 14]).

The *RSA trapdoor one-way function* is just the discrete exponentiation

$$f_z(x) = x^e \pmod{n} \tag{40}$$

where x is a nonnegative integer less than $n = pq$ and where the "trapdoor" $z = \{p, q, e\}$; here p and q are distinct very large primes such that $\phi(n) = (p - 1)(q - 1)$ has a very large prime factor, and e is a positive integer less than $\phi(n)$ such that GCD(e, $\phi(n)$) = 1. The easy-to-find algorithm E_z to compute f_z easily is exponentiation by square-and-multiply; *publishing this algorithm amounts just to publishing n and e.* The inverse function is

$$f_z^{-1} = y^d \pmod{n} \tag{41}$$

where d is the unique positive integer less than n such that

$$de = 1 \pmod{\phi(n)}. \tag{42}$$

The easy-to-find (when one knows z) algorithm D_z to compute f_z^{-1} is also exponentiation by square-and-multiply; the decrypting exponent d is found using Euclid's algorithm for computing GCD(e, $\phi(n)$).

Note that the domain and range of the RSA trapdoor one-way function coincide, both are the set of integers from 0 to $m - 1$ inclusive. This means that the RSA function can be used to form digital signatures in the manner suggested by Diffie and Hellman. This digital signature capability is one of the most important and useful features of the RSA function.

That Eq. (41) really gives the inverse function for Eq. (40) can be seen as follows. Equation (42) is equivalent to the statement (in ordinary integer arithmetic) that

$$de = \phi(n)Q + 1 \tag{43}$$

for some integer Q. From Eqs. (40) and (43), we obtain

$$\begin{aligned}(x^e)^d &= x^{\phi(n)Q+1} && \pmod{n}\\ &= (x^{\phi(n)})^Q x && \pmod{n}\\ &= x && \pmod{n}\end{aligned} \tag{44}$$

where at the last step we used Euler's theorem [Eq. (38)]. [The wary reader will have noted that Euler's theorem requires GCD(x, n) = 1; but in fact Eq. (44) holds for all nonnegative integers x less than n in the special case when n is the product of two distinct primes.] Equation (44) shows that raising a number to the d power (mod n) is indeed the inverse of raising a number to the e power (mod n). It remains to show why RSA believed (as do most cryptographers today) that it is computationally impossible to invert this function f_z when one knows only n and e, and also how it is possible easily to choose *randomly* the two distinct and very large primes p and q, as must be done for an enemy to be unable to guess p and q.

The enemy knows only n and e. But if he can factor $n = pq$, then he knows the entire trapdoor $z = \{p, q, e\}$, and hence can decrypt just as readily as the legitimate receiver. The security of the RSA public key cryptosystem depends on the assumption that *any way of inverting* f_z is equivalent *to factoring* $n = pq$, that is, given any way to invert f_z, one could with at most a little more computational work go on to factor n. In their paper [25], RSA show that this is true for the most likely ways that one might try to factor n, but the assumption has never been proved. But is the attack by factoring n computationally infeasible? The answer is yes if one chooses p and q on the order of 100 decimal digits each (as RSA suggested thirteen years ago) *and* if there is no revolutionary breakthrough in factoring algorithms. As Rivest [27] recently pointed out, all of the best factoring algorithms today have running times upper-bounded by the same peculiar-looking function which, for numbers to be factored between 50 and 200 decimal digits, increases by a factor of 10 for every additional 15 digits (roughly) in the number. Today it takes about 1 day on a supercomputer to factor a number of about 80 decimal digits. It would take 10^8 times that long to factor a 200 digit number $n = pq$, roughly half a million years! One of the by-products of the RSA paper has been a revival of interest in factoring, but this accelerated research effort has produced no revolutionary breakthrough. Proponents of the RSA public key cryptosystem believe that it never will. An interesting fact is that the best algorithms today for solving the (mod p) discrete logarithm problem [28] and the best algorithms for factoring n [29] require a computational effort that grows asymptotically in the same manner with p and n, respectively. Thus the RSA trapdoor function [Eq. (49)] and the Diffie–Hellman function [Eq. (33)] have, as of today, about the same claim to be called "one-way." For given $n \approx p$, however, the Diffie–Hellman function appears more difficult to invert.

It remains to consider how one can randomly choose the very large primes, p and q, required for RSA. A theorem of Tchebychef, (see [30, pp. 9–10]), states that the fraction of positive integers less than any large integer m that are primes is close to

$(\ln m)^{-1}$. For instance, the fraction of integers less than 10^{100} that are primes is about $(\ln 10^{100})^{-1} \approx \frac{1}{230}$. Because 90 percent of these integers lie between 10^{99} and 10^{100}, the fraction of primes in this range is also about $\frac{1}{230}$. Thus, if one chooses an integer between 10^{99} and 10^{100} completely at random, the chances that one chooses a prime are about $\frac{1}{230}$. One easily doubles the odds to $\frac{1}{115}$ if one is sensible enough to choose only odd integers. One needs then only about 115 such choices on the average before one has chosen a prime. But how does one recognize a prime? It is a curious fact that one can rather easily test quite reliably whether an integer is a prime or not, even if one cannot factor that integer after one discovers that it is not a prime. Such primality tests rely on a *theorem of Fermat* (1601–1665) that asserts that for any positive integer b less than a prime p

$$b^{p-1} = 1 \quad (\text{mod } p) \tag{45}$$

For instance, $2^4 = 1$ (mod 5). [The reader may have noticed that Eq. (45) is a special case of Eq. (38), but should remember that Fermat lived a century before Euler!] If one has an integer r that one wishes to test for primeness, one can choose any positive integer b less than r and check whether

$$b^{r-1} \overset{?}{=} 1 \quad (\text{mod } r) \tag{46}$$

If the answer is no, one has the absolute assurance of Fermat that r is not a prime. If the answer is yes, one can begin to suspect that r is a prime, and one then christens r a *pseudoprime* to the base b. If r is not a prime, it turns out that it can be a pseudoprime for fewer (actually many fewer) than about half of the possible bases b [except for the very rare Carmichael numbers, which are non-primes r that pass Fermat's test for every base b relatively prime to r, but are detected by a slight extension of Fermat's test]. Thus if r is very large, and one independently chooses t bases b completely at random, the probability is less than about 2^{-t} that r will pass Fermat's test [Eq. (46)] for all these bases if r is not truly a prime. If we take, say $t = 100$, then we can be virtually certain that r is a prime if it passes t indepedent Fermat tests. Such "probabilistic tests for primeness" were introduced by Solovay and Strassen, and have been further refined by Rabin [31]. Such tests are today being used to check randomly chosen odd integers for primeness until one has found the two distinct large primes one needs for the RSA trapdoor one-way function, or, more precisely, until one is sufficiently sure that he has found two such primes.

The technique just described leads to the formation of large randomly chosen "probable primes" and is the technique currently in widest use for finding the large primes needed with RSA. There is an alternative approach, however, that leads to *sure* primes that are "probably randomly chosen." It is not hard to "grow" large primes with a probabilistic algorithm; the trick is to make the primes appear to be chosen according to a probability distribution that is as uniform as possible over some interval. Maurer [54] has recently given such an algorithm that is very fast (its running time is about the same as for $t = 4$ Fermat tests) and plausibly gives an almost uniform distribution for the selected primes. It would not be surprising should this or similar algorithms eventually replace prime-testing as the method of choice for finding the large primes needed in the RSA public key cryptosystem or in the Diffie–Hellman public key-distribution system.

There are very large-scale integration (VLSI) chips today that can implement the RSA encrypting and decrypting function at a data rate of a few kilobits per second. (These same chips can also be used to implement Fermat's test, and thus to find the needed 100 decimal digit primes, p and q). Rivest [27] has given convincing arguments that significantly higher data rates will never be achieved. For many cryptographic applications, these data rates are too low. In such cases, the RSA public key cryptosystem may still desirably be used to distribute the secret keys that will then be used in high-speed secret key ciphers, such as DES or certain stream ciphers. And the RSA algorithm may still desirably be used for authentication in its "digital signature" mode.

Before closing this section on the RSA system, we should mention that Rabin [32] has developed a variant of the RSA public key system for which he *proved* that being able to find the plaintext X from the cryptogram Y is *equivalent to factoring* $n = pq$. The system is somewhat more complicated than basic RSA, but Williams [33] refined the variant so that the extra complication is quite tolerable. This might seem to be the ultimate "RSA system," but paradoxically the breaking-is-provably-equivalent-to-factoring versions of RSA have a new weakness that was pointed out by Rivest. The proof of their equivalence to factoring is *constructive*, that is, one shows that if one could solve $Y = X^e \pmod{n}$ for X in these systems [which differ from RSA in that now $\text{GCD}(e, \phi(n)) \neq 1$], then one could easily go on to factor X. But this means that these systems *succumb to a chosen-ciphertext attack* in which an enemy randomly chooses X', computes $Y = (X')^e$ and then submits Y to the decrypter, who returns a solution X of $Y = X^e$ [where the fact that $\text{GCD}(e, \phi(n)) \neq 1$ results in the situation that the solution is not unique so that $X \neq X'$ is possible]. The chances are $\frac{1}{2}$ that the returned X together with X' will give the enemy the information necessary to factor $n = pq$ and thus to break the system. In a public key environment, such a chosen-ciphertext attack becomes a distinct possibility. The net result is that most cryptographers prefer to use the original RSA public key cryptosystem, and to pray for the day when a *nonconstructive* proof is given that breaking it is equivalent to factoring.

This is perhaps the appropriate point to mention that a *public key cryptosystem, if it is secure at all, is secure against a chosen-plaintext attack.* For the enemy cryptanalyst is always welcome to fetch the algorithm E_z from the public directory and then to compute the cryptograms, $y = f_y(x)$, for as many plaintexts x as he pleases. This shows that a trapdoor one-way function must necessarily be much more difficult to invert than the encrypting function of a conventional secret key cipher that is also secure against a chosen-plaintext attack. In the latter case, the enemy can still (by assumption) obtain the cryptograms y, for whatever plaintexts x, he pleases. But the enemy no longer has the luxury of watching the encryption algorithm execute its encryptions, because the secret key is an ingredient of the algorithm.

3.4 Some Remarks on Public Key Cryptography

The Diffie–Hellman one-way function and the RSA trapdoor one-way function suffice to illustrate the main ideas of public key cryptography, which is why we have given them rather much attention. But a myriad of other such functions have been proposed. Some have almost immediately been exposed as insecure, others appear promising. But no one has yet produced a proof that any function is a one-way function or a trapdoor one-way function. Even the security of the Rabin variant of RSA rests on the unproved (but very plausible) assumption that factoring large integers is computationally infeasible.

There has been some hope that the new, but rapidly evolving, theory of computational complexity, particularly Cook and Karp's theory of nondeterministic-polynomial (NP) completeness, (see [34]), will lead to provably one-way functions or provably trapdoor one-way functions. This hope was first expressed by Diffie and Hellman [12], but has thus far led mainly to failures such as the spectacular failure of the Merkle–Hellman trapdoor-knapsack public key cryptosystem. Part of the difficulty has been that NP-completeness is a worst-case phenomenon, not a "virtually all cases" phenomenon as one requires in public key cryptography. For instance, Even, Lempel, and Yacobi have constructed an amusing example of a public key cryptosystem whose breaking is equivalent to solving an "NP-hard" problem, but which can virtually always be broken [35]. [A problem is NP-hard if its solution is at least as difficult as the solution of an NP-complete problem.] But the greater difficulty has been to formulate a trapdoor one-way function whose inversion would require the solution of an NP-complete problem; this has not yet been accomplished. For instance, the inversion of the Merkle–Hellman trapdoor-knapsack one-way function is actually an easy problem disguised to resemble an NP-hard problem; Shamir broke this public key cipher, not by solving the NP-hard problem, but by stripping off the disguise.

We are grateful to J. Denés for calling our attention to the fact that the notion of "one-wayness" is much older than we had suspected. W. S. Jevons, in his book [55] first published in 1873, wrote:

> There are many cases in which we can easily and infallibly do a certain thing but may have much trouble in undoing it. . . . Given any two numbers, we may by a simple and infallible process obtain their product, but when a large number is given it is quite another matter to determine its factors. Can the reader say what two numbers multiplied together will produce the number 8 616 460 799? I think it is unlikely that anyone but myself will ever know; for they are *two large prime numbers* (emphasis added).

Thirty years later, Lehmer [56] announced the "two large prime numbers" to be 89 681 and 96 079, but added "I think that the number has been resolved before, but I do not know by whom." Such anecdotes as that just recounted here serve to feed the suspicions of those who innately mistrust public key cryptography and who will continue to do so until a provably secure public key cipher is produced. But, as we have stressed above, the security of all known practical secret key ciphers also rests on conjectures. Neither the secret key advocate nor the public key advocate is in a good position to hurl stones at the other.

4 CRYPTOGRAPHIC PROTOCOLS

4.1 What Is a Protocol?

It is difficult to give a definition of "protocol" that is both precise and general enough to encompass most things to which people apply this label in cryptography and elsewhere. Roughly speaking, we might say that a *protocol* is a multiparty algorithm, that is, a specified sequence of actions by which two or more parties cooperatively accomplish some task. Sending a secret message from one user to another in a large network by means of a public key cryptosystem, for instance, can be considered a protocol, based on a trapdoor one-way function, by means of which the users of the system and the custodian of the public directory cooperate to ensure the privacy of messages sent from one user to another.

4.2 A Key-Distribution Protocol

Many cryptographers, particularly those skeptical of public key ideas, consider the *key management problem* (that is, the problem of securely distributing and changing secret keys) to be the main practical problem in cryptography. For example, if there are S users in the system, one will need $S(S - 1)/2$ different secret keys if one is to have a dedicated secret key for every possible pair of users—an unwelcome prospect in a large system. It is unlikely that any user will ever wish to send secret messages to more than a few other users, but in advance one usually does not know who will later want to talk secretly to whom. A popular solution to this problem is the following key-distribution protocol that requires the advance distribution of only S secret keys, but still permits any pair of users to communicate secretly; there is a needed new entity, however, the *trusted key distribution center* (TKDC).

Key-Distribution Protocol

1. The TKDC securely delivers a randomly chosen secret key Z_i to user i in the system, for $i = 1, 2, \ldots, S$.

2. When user i wishes to communicate secretly to user j, he sends the TKDC a request (which can be in the clear) over the public network for a secret key to be used for this communication.

3. The TKDC randomly chooses a new secret key Z_{ij} which it treats as part of the plaintext. The other part of the plaintext is a "header" in which user i and user j are identified. The TKDC encrypts this plaintext in both key Z_i and key Z_j with whatever secret key cipher is installed in the system, then sends the first cryptogram to user i and the second to user j over the public network.

4. Users i and j decrypt the cryptograms they have just received and thereby obtain the secret key to be used for encrypting further messages between these two users.

This protocol sounds innocent enough, but its security against a ciphertext-only attack requires more than ciphertext-only security of the system's secret key cipher. Why? Because in step (3) we see that an enemy cryptanalyst will have access to two cryptograms in different keys for the *same* plaintext. This can be helpful to the cryptanalyst, although it does not give him as much information as he could get in a chosen-plaintext attack on the individual ciphers. Thus, security of the system's cipher against a chosen-plaintext attack will make this protocol also secure against chosen-plaintext attacks. The point to be made here is that when one embeds a cipher into a protocol, *one must be very careful to ensure that whatever security is assumed for the cipher is not compromised by the protocol.*

4.3 Shamir's Three-Pass Protocol

One of the most interesting cryptographic protocols, due to A. Shamir in unpublished work, shows that secrecy can be obtained with no advance distribution of either secret keys or public keys. The protocol assumes two users connected by a link (such as a seamless optical fiber or a trustworthy but curious postman) that guarantees that the enemy cannot insert, or tamper with, messages but allows the enemy to read all messages sent over the link. The users are assumed to have a secret key cipher system whose encrypting function $E_z(\cdot)$ has the *commutative property,* that, for all plaintexts, x, and all keys, z_1 and z_2,

$$E_{z_2}(E_{z_1}(x)) = E_{z_1}(E_{z_2}(x)) \qquad (47)$$

that is, the result of a double encryption is the same whether one uses first the key z_1 and then the key z_2 or vice versa. There are many such ciphers, for example, the one-time pad [Eq. (4)] fits the bill because $(x \oplus z_1) \oplus z_2 = (x \oplus z_2) \oplus z_1$, where the addition is bit-by-bit mod 2.

Shamir's Three-Pass Protocol

1. Users A and B randomly choose their own private secret keys, Z_A and Z_B, respectively.

2. When user A wishes to send a secret message X to user B, user A encrypts X with his own key Z_A and sends the resulting cryptogram $Y_1 = E_{Z_A}(X)$ on the open-but-tamperproof link to user B.

3. User B, upon receipt of Y_1, treats Y_1 as plaintext and encrypts Y_1 with his own key Z_B and sends the resulting cryptogram $Y_2 = E_{Z_B}(Y_1) = E_{Z_B}(E_{Z_A}(X))$ on the open-but-tamperproof link to user A.

4. User A, upon receipt of Y_2, decrypts Y_2 with his own key Z_A. Because of the commutative property [Eq. (47)], this removes the former encryption by Z_A and results in $Y_3 = E_{Z_B}(X)$. User A then sends Y_3 over the open-but-tamper-proof link to user B.

5. User B, upon receipt of Y_3, decrypts Y_3 with his own key Z_B to obtain X, the message that A has now successfully sent to him secretly.

What secret key cipher shall we use in this protocol? Why not the one-time pad, a cipher that gives perfect secrecy? If we use the one-time pad, the three cryptograms become

$$\begin{aligned} Y_1 &= X \oplus Z_A \\ Y_2 &= X \oplus Z_A \oplus Z_B \\ Y_3 &= X \oplus Z_B \end{aligned} \qquad (48)$$

The enemy cryptanalyst sees all three cryptograms, and hence can form

$$Y_1 \oplus Y_2 \oplus Y_3 = X$$

where we have used the fact that two identical quantities sum to O mod 2. Thus, the three-pass protocol is completely insecure when we use the one-time pad for the embedded cipher! The reason for this is, as Eq. (48) shows, that the effect of the protocol is that each of the two ciphers get used "$1\frac{1}{2}$ times," rather than only once as is required for the security of the "one-time" pad.

Is there a cipher that can be used in the Shamir three-pass protocol and still retain its security? There seems to be. Let p be any large prime for which $p - 1$ has a large prime factor (to make the discrete logarithm problem in mod p arithmetic computationally infeasible to solve). Randomly choose a positive integer e less than $p - 1$ such that GCD$(e, p - 1) = 1$, and let d be the unique positive integer less than $p - 1$ such that

$$de = 1 \pmod{p - 1} \qquad (49)$$

Let $Z = (d, e)$ be the secret key and take the encrypting and decrypting functions to be

$$\begin{aligned} y &= E_z(x) = x^e \pmod{p} \\ x &= D_z(y) = y^d \pmod{p} \end{aligned} \qquad (50)$$

where x and y are positive integers less than p. [The fact that $y^d = x^{de} = x \pmod{p}$ is an easy consequence of Fermat's theorem [Eq. (45)] and the fact that Eq. (49) implies

$de = Q(p - 1) + 1$ for some integer Q.] That this cipher has the commutative property in Eq. (47) follows from Eq. (50) because

$$(x^{e_1})^{e_2} = x^{e_1e_2} = (x^{e_2})^{e_1} \pmod p$$

When this cipher is used in the three-pass protocol, the three cryptograms become

$$\begin{aligned} y_1 &= x^{e_A} \pmod p \\ y_2 &= x^{e_Ae_B} \pmod p \\ y_3 &= x^{e_B} \pmod p \end{aligned} \tag{51}$$

If one can solve the discrete logarithm problem, one can obtain

$$\log_\alpha y_1 = e_A \log_\alpha x \pmod{p-1} \tag{52a}$$

$$\log_\alpha y_2 = e_Ae_B \log_\alpha x \pmod{p-1} \tag{52b}$$

where α is any chosen primitive element for arithmetic mod p, and where we have used the fact that the arithmetic of discrete logarithms is mod $(p - 1)$ arithmetic—this follows from Fermat's theorem [Eq. (45)] that gives $\alpha^{p-1} = 1 = \alpha^0$. We can now use Euclid's extended GCD algorithm (see Section 3.3) to find the positive integer b less than $p - 1$ such that

$$b \log_\alpha y_1 = 1 \pmod{p-1}$$

which from Eq. (52a) further implies

$$be_A \log_\alpha x = 1 \pmod{p-1} \tag{53}$$

Multiplying Eq. (52b) by b on both sides, then using Eq. (53), we obtain

$$b \log_\alpha y_2 = e_B \pmod{p-1} \tag{54}$$

Thus, an enemy who can solve the discrete logarithm problem for modulo-p arithmetic can find e_B, hence also d_B, and thus read the message x just as well as user B. There seems to be no way for the enemy to find x without equivalently solving the discrete logarithm problem, but (like so many other things in public key cryptography) this has never been proved. This particular cipher for the three-pass protocol was proposed by Shamir (and independently but later by J. Omura, who was aware of Shamir's three-pass protocol, but unaware of his proposed cipher for the protocol).

4.4 Conclusion

There are many protocols that have been proposed recently by cryptologic researchers. One of the most amusing is the Shamir–Rivest–Adleman protocol for ''mental poker,'' a protocol that manages to allow an honest game of poker to be played with no cards [36]. Such frivolous-sounding protocols have a serious cryptographic purpose; however, in this case one could take the purpose to be a protocol for assuring the authenticity of randomly chosen numbers. Similarly, Chaum [37] has proposed an interesting protocol by which parties making transactions through a bank can do so without the bank ever knowing who is paying what to whom that also suggests a cryptographic application in key distribution. Protocol formulation has recently gained new momentum and has become one of the most active areas of current cryptologic research, as well as one of the most difficult, particularly when one seeks particular cryptographic functions to imbed in the protocol without compromise of their security. The RSA trapdoor one-way function is far and away the most frequently used function for this purpose.

We have not mentioned many of the important contributions to cryptology made in the past 10 years. It has *not* been our purpose to *survey* research in cryptology, but rather to sketch the intellectual outlines of the subject. Readers wishing to keep abreast of current research in cryptology will find the Proceedings of the CRYPTO conference (held annually in Santa Barbara, California since 1981) and of the EUROCRYPT conference (held annually since 1982) to be invaluable. There are also several journals either completely dedicated to the field, such as the *Journal of Cryptology* or *Cryptologia,* or else with special emphasis on the subject, such as *Designs, Codes and Cryptography* or the *IEEE Transactions on Information Theory.* In addition several recent general textbooks [38]–[42], [58], [59] on cryptology will give the reader an orderly development of the subject. Two recent texts [43], [57] give a broad treatment of the number-theoretic concepts on which much of present-day public key cryptology depends. The book by Rueppel [44] is a good source of information about stream ciphers.

REFERENCES

[1] W. Diffie and M. E. Hellman, ''Privacy and authentication: An introduction to cryptography,'' *Proc. IEEE,* vol. 67, pp. 397–427, March 1979.

[2] G. J. Simmons, ''Cryptology,'' in *Encyclopaedia Britannica,* ed. 16. Chicago: Encyclopaedia Britannica Inc., 1986, pp. 913–924B.

[3] D. Kahn, *The Codebreakers, The Story of Secret Writing.* New York: Macmillan, 1967.

[4] ——— , *The Codebreakers, The Story of Secret Writing,* abridged ed. New York: New American Library, 1973.

[5] R. C. Merkle and M. E. Hellman, ''Hiding information and signatures in trapdoor knapsacks,'' *IEEE Trans. Informat. Theory,* vol. IT-24, pp. 525–530, Sept. 1978.

[6] A. Shamir, ''A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem,'' *IEEE Trans. Informat. Theory,* vol. IT-30, pp. 699–704, Sept. 1984.

[7] D. B. Newman, Jr., and R. L. Pickholtz, ''Cryptography in the private sector,'' *IEEE Commun. Mag.,* vol. 24, pp. 7–10, Aug. 1986.

[8] G. S. Vernam, ''Cipher printing telegraph systems for secret wire and radio telegraphic communications,'' *J. Am. Inst. Elec. Eng.,* vol. 55, pp. 109–115, 1926.

[9] A. Hodges, *Alan Turing, The Enigma.* New York: Simon and Schuster, 1983.

[10] C. E. Shannon, ''Communication theory of secrecy systems,'' *Bell Syst. Tech. J.,* vol. 28, pp. 656–715, Oct. 1949.

[11] ——— , ''A mathematical theory of communication,'' *Bell Syst. Tech. J.,* vol. 27, pp. 379–423, 623–656, July and Oct. 1948.

[12] W. Diffie and M. E. Hellman, ''New directions in cryptography,'' *IEEE Trans. Informat. Theory,* vol. IT-22, pp. 644–654, Nov. 1976.

[13] R. C. Merkle, "Secure communication over insecure channels," *Comm. ACM,* vol. 21, pp. 294–299, Apr. 1978.

[14] G. J. Simmons, "Authentication theory/coding theory," in *Advances in Cryptology, Proceedings of CRYPTO 84,* G. R. Blakley and D. Chaum, Eds. Lecture Notes in Computer Science, No. 196. Berlin: Springer-Verlag, 1985, pp. 411–431.

[15] W. Feller, *An Introduction to Probability Theory and Its Applications,* vol. 2. New York: Wiley, 1966.

[16] "Data encryption standard," Federal Information Processing Standard PUB 46, National Tech. Info. Service, Springfield, VA, 1977.

[17] R. Morris, "The data encryption standard—retrospective and prospects," *IEEE Commun. Mag.*, vol. 16, pp. 11–14, Nov. 1978.

[18] W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer,* vol. 10, pp. 74–84, June 1977.

[19] M. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Trans. Informat. Theory,* vol. IT-26, pp. 401–406, July 1980.

[20] R. C. Merkle and M. E. Hellman, "On the security of multiple encryption," *Comm. ACM,* vol. 24, pp. 465–467, July 1981.

[21] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Informat. Theory,* vol. IT-15, pp. 122–127, Jan. 1969.

[22] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Trans. Informat. Theory,* vol. IT-30, pp. 776–780, Sept. 1984.

[23] J. L. Massey and I. Ingemarsson, "The Rip van Winkle cipher—A simple and provably computationally secure cipher with a finite key," in *IEEE Int. Symp. on Informat. Theory,* (Brighton, England) (abstr.), p. 146, June 24–28, 1985.

[24] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms in GF(p) and its cryptographic significance," *IEEE Trans. Informat. Theory,* vol. IT-24, pp. 106–110, Jan. 1978.

[25] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. ACM,* vol. 21, pp 120–126, Feb. 1978.

[26] D. E. Knuth, *The Art of Computer Programming.* vol. 1. *Fundamental Algorithms.* Reading, MA: Addison-Wesley, 1973.

[27] R. L. Rivest, "RSA chips (past/present/future)," presented at Eurocrypt 84, Paris, Apr. 9–11, 1984.

[28] A. M. Odlyzko, "On the complexity of computing discrete logarithms and factoring integers," in *Open Problems in Communication and Computation,* T. M. Cover and B. Gopinath, Eds. New York: Springer, pp. 113–116, 1987.

[29] C. Pomerance, "Analysis and comparison of some integer factoring algorithms," in *Computational Number Theory,* H. W. Lenstra, Jr., and R. Tijdeman, Eds. Amsterdam, The Netherlands: Mathematics Centre Tract, 1982.

[30] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers,* ed. 4. London: Oxford, 1960.

[31] M. O. Rabin, "Probabilistic algorithm for primality testing," *J. Number Theory,* vol. 12, pp. 128–138, 1980.

[32] ——— , "Digital signatures and public-key functions as intractable as factorization," Tech. Rep. LCS/TR212, Massachusetts Institute of Technology Laboratory for Computer Science, Cambridge, MA, 1979.

[33] H. C. Williams, ''An M^3 public-key encryption scheme,'' in *Advances in Cryptology, Proceedings of CRYPTO 85*, H. C. Williams, Ed. Lecture Notes in Computer Science, No. 218. Berlin: Springer-Verlag, 1985, pp. 358–368.

[34] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman, 1979.

[35] A. Lempel, ''Cryptology in transition,'' *Computing Surv.*, vol. 11, pp. 285–303, Dec. 1979.

[36] A. Shamir, R. L. Rivest, and L. Adleman, ''Mental poker,'' in *Mathematical Gardener*, D. E. Klarner, Ed. New York: Wadsworth, 1981, pp. 37–43.

[37] D. Chaum, ''Security without identification: Transaction systems to make big brother obsolete,'' *Comm. ACM*, vol. 28, pp. 1030–1044, Oct. 1985.

[38] H. Beker and F. Piper, *Cipher Systems, The Protection of Communications*. London: Northwood Books, 1982.

[39] D. W. Davies and W. L. Price, *Security for Computer Networks*. New York: Wiley, 1984.

[40] D. E. R. Denning, *Cryptography and Data Security*. Reading, MA: Addison-Wesley, 1982.

[41] A. C. Konheim, *Cryptography, A Primer*. New York: Wiley, 1981.

[42] C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Computer Data Security*. New York: Wiley, 1982.

[43] E. Kranakis, *Primality and Cryptography*. New York: Wiley, 1986.

[44] R. Rueppel, *Analysis and Design of Stream Ciphers*. New York: Springer, 1986.

[45] J. L. Massey, ''An introduction to contemporary cryptology,'' *Proc. IEEE*, vol. 76, pp. 533–549, May 1988.

[46] L. Kruh, ''Stimson, the black chamber, and the gentlemen's mail quote,'' *Cryptologia*, vol. 12, pp. 65–89, Apr. 1988.

[47] H. L. Stimson and McG. Bundy, *On Active Service in Peace and War*. New York: Harper & Bros., 1947.

[48] C. G. Günther, ''A universal algorithm for homophonic coding,'' in *Advances in Cryptology—Eurocrypt'88*, C. G. Gunther, Ed. Lecture Notes in Computer Science, No. 330. Berlin: Springer-Verlag, 1988.

[49] H. N. Jendahl, Y. J. B. Kuhn, and J. L. Massey, ''An information-theoretic treatment of homophonic substitution,'' in *Advances in Cryptology—Eurocrypt'89*, J. -J. Quisquater and J. Vandewalle, Eds. Lecture Notes in Computer Science, No. 434. Berlin: Springer-Verlag, 1990, pp. 382–394.

[50] A. Sgarro, ''Informational divergence bounds for authentication codes,'' in *Advances in Cryptology—Eurocrypt'89*, J. -J. Quisquater and J. Vandewalle, Eds. Lecture Notes in Computer Science No.434. Berlin: Springer-Verlag, 1990, pp. 93–101.

[51] R. Johannesson and A. Sgarro, ''Strengthening Simmons' bound on impersonation,'' *IEEE Trans. Informat. Theory*, vol. 37, no. 4, pp. 1182–1185, July 1991.

[52] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. New York: Academic Press, 1981.

[53] U. M. Maurer, ''A provably-secure strongly-randomized cipher,'' in *Advances in Cryptology—Eurocrypt'90*, I. Dåmgard, Ed. Lecture Notes in Computer Science. New York and Heidelberg: Springer-Verlag, no. 473, pp. 361–373, 1991.

[54] U. M. Maurer, ''Fast Generation of secure RSA-moduli with almost maximum diversity,'' in *Advances in Cryptology—Eurocrypt'89*, J. -J. Quisquater and J.

Vandewalle, Eds. Lecture Notes in Computer Science, No. 434. Berlin: Springer-Verlag, 1990, pp. 636–647.

[55] W. S. Jevons, *The Principles of Science* (1st ed. 1873, 2nd ed. 1883). New York: Dover, 1958.

[56] D. H. Lehmer, ‘‘A theorem in the theory of numbers,’’ *Bull. Amer. Math. Soc.*, Vol.13, no.2, pp. 501–502, July 1907.

[57] N. Koblitz, *A Course in Number Theory and Cryptography.* New York: Springer, 1987.

[58] H. C. A. van Tilborg, *An Introduction to Cryptology.* Norwell, MA: Kluwer Academic, 1988.

[59] J. Seberry and J. Pieprzyk, *Cryptography: An Introduction to Computer Security.* Englewood Cliffs, NJ: Prentice Hall, 1988.