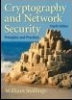


**Cryptography and Network Security Principles and Practices, Fourth Edition**

****

• Table of Contents

• Index

By William Stallings

**Publisher**: **Prentice Hall**

**Pub Date**: **November 16, 2005**

**Print ISBN-10**: **0-13-187316-4**

**Print ISBN-13**: **978-0-13-187316-2**

**eText ISBN-10**: **0-13-187319-9**

**eText ISBN-13**: **978-0-13-187319-3**

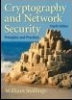
**Pages**: **592**

In this age of viruses and hackers, of electronic eavesdropping and electronic fraud, security is paramount.

As the disciplines of cryptography and network security have matured, more practical, readily available applications to enforce network security have developed. This text provides a practical survey of both the principles and practice of cryptography and network security. First, the basic issues to be addressed by a network security capability are explored through a tutorial and survey of cryptography and network security technology. Then, the practice of network security is explored via practical applications that have been implemented and are in use today.





**Cryptography and Network Security Principles and Practices, Fourth Edition** 

By William Stallings

**Publisher**: **Prentice Hall**

**Pub Date**: **November 16, 2005**

**Print ISBN-10**: **0-13-187316-4**

**Print ISBN-13**: **978-0-13-187316-2**

• Table of Contents • Index

Copyright

**eText ISBN-10**: **0-13-187319-9**

**eText ISBN-13**: **978-0-13-187319-3**

**Pages**: **592**

Notation xi Preface xiii Objectives xiii Intended Audience xiii Plan of the Book xiv Internet Services for Instructors and Students xiv Projects for Teaching Cryptography and Network Security xiv What's New in the Fourth Edition xv Acknowledgments xvi

Chapter 0. Reader's Guide 1 Section 0.1. Outline of this Book 2 Section 0.2. Roadmap 2 Section 0.3. Internet and Web Resources 4 Chapter 1. Introduction 6 Section 1.1. Security Trends 9 Section 1.2. The OSI Security Architecture 12

Section 1.3. Security Attacks 13 Section 1.4. Security Services 16 Section 1.5. Security Mechanisms 19 Section 1.6. A Model for Network Security 22 Section 1.7. Recommended Reading and Web Sites 24 Section 1.8. Key Terms, Review Questions, and Problems 25 Part One: Symmetric Ciphers 26 Chapter 2. Classical Encryption Techniques 28 Section 2.1. Symmetric Cipher Model 30

Section 2.2. Substitution Techniques 35 Section 2.3. Transposition Techniques 49 Section 2.4. Rotor Machines 51 Section 2.5. Steganography 53 Section 2.6. Recommended Reading and Web Sites 55 Section 2.7. Key Terms, Review Questions, and Problems 56 Chapter 3. Block Ciphers and the Data Encryption Standard 62 Section 3.1. Block Cipher Principles 64 Section 3.2. The Data Encryption Standard 72 Section 3.3. The Strength of Des 82 Section 3.4. Differential and Linear Cryptanalysis 83 Section 3.5. Block Cipher Design Principles 86 Section 3.6. Recommended Reading 90 Section 3.7. Key Terms, Review Questions, and Problems 90 Chapter 4. Finite Fields 95 Section 4.1. Groups, Rings, and Fields 97 Section 4.2. Modular Arithmetic 101 Section 4.3. The Euclidean Algorithm 107

Section 4.4. Finite Fields of The Form GF(p) 109 Section 4.5. Polynomial Arithmetic 113 Section 4.6. Finite Fields Of the Form GF(2n) 119 Section 4.7. Recommended Reading and Web Sites 129 Section 4.8. Key Terms, Review Questions, and Problems 130 Chapter 5. Advanced Encryption Standard 134 Section 5.1. Evaluation Criteria For AES 135 Section 5.2. The AES Cipher 140 Section 5.3. Recommended Reading and Web Sites 160 Section 5.4. Key Terms, Review Questions, and Problems 161 Appendix 5A Polynomials with Coefficients in GF(28) 163 Appendix 5B Simplified AES 165 Chapter 6. More on Symmetric Ciphers 174 Section 6.1. Multiple Encryption and Triple DES 175 Section 6.2. Block Cipher Modes of Operation 181 Section 6.3. Stream Ciphers and RC4 189 Section 6.4. Recommended Reading and Web Site 194 Section 6.5. Key Terms, Review Questions, and Problems 194 Chapter 7. Confidentiality Using Symmetric Encryption 199 Section 7.1. Placement of Encryption Function 201 Section 7.2. Traffic Confidentiality 209

Section 7.3. Key Distribution 210 Section 7.4. Random Number Generation 218 Section 7.5. Recommended Reading and Web Sites 227 Section 7.6. Key Terms, Review Questions, and Problems 228 Part Two: Public-Key Encryption and Hash Functions 232 Chapter 8. Introduction to Number Theory 234 Section 8.1. Prime Numbers 236 Section 8.2. Fermat's and Euler's Theorems 238 Section 8.3. Testing for Primality 242 Section 8.4. The Chinese Remainder Theorem 245 Section 8.5. Discrete Logarithms 247 Section 8.6. Recommended Reading and Web Sites 253 Section 8.7. Key Terms, Review Questions, and Problems 254 Chapter 9. Public-Key Cryptography and RSA 257 Section 9.1. Principles of Public-Key Cryptosystems 259 Section 9.2. The RSA Algorithm 268 Section 9.3. Recommended Reading and Web Sites 280 Section 9.4. Key Terms, Review Questions, and Problems 281 Appendix 9A Proof of the RSA Algorithm 285 Appendix 9B The Complexity of Algorithms 286 Chapter 10. Key Management; Other Public-Key Cryptosystems 289 Section 10.1. Key Management 290 Section 10.2. Diffie-Hellman Key Exchange 298 Section 10.3. Elliptic Curve Arithmetic 301 Section 10.4. Elliptic Curve Cryptography 310 Section 10.5. Recommended Reading and Web Sites 313 Section 10.6. Key Terms, Review Questions, and Problems 314 Chapter 11. Message Authentication and Hash Functions 317 Section 11.1. Authentication Requirements 319 Section 11.2. Authentication Functions 320 Section 11.3. Message Authentication Codes 331 Section 11.4. Hash Functions 334 Section 11.5. Security of Hash Functions and Macs 340 Section 11.6. Recommended Reading 344 Section 11.7. Key Terms, Review Questions, and Problems 344 Appendix 11A Mathematical Basis of the Birthday Attack 346 Chapter 12. Hash and MAC Algorithms 351 Section 12.1. Secure Hash Algorithm 353 Section 12.2. Whirlpool 358

Section 12.3. HMAC 368 Section 12.4. CMAC 372 Section 12.5. Recommended Reading and Web Sites 374 Section 12.6. Key Terms, Review Questions, and Problems 374 Chapter 13. Digital Signatures and Authentication Protocols 377 Section 13.1. Digital Signatures 378 Section 13.2. Authentication Protocols 382 Section 13.3. Digital Signature Standard 390 Section 13.4. Recommended Reading and Web Sites 393 Section 13.5. Key Terms, Review Questions, and Problems 393 Part Three: Network Security Applications 398 Chapter 14. Authentication Applications 400 Section 14.1. Kerberos 401 Section 14.2. X.509 Authentication Service 419 Section 14.3. Public-Key Infrastructure 428 Section 14.4. Recommended Reading and Web Sites 430 Section 14.5. Key Terms, Review Questions, and Problems 431 Appendix 14A Kerberos Encryption Techniques 433 Chapter 15. Electronic Mail Security 436 Section 15.1. Pretty Good Privacy 438 Section 15.2. S/MIME 457 Section 15.3. Key Terms, Review Questions, and Problems 474 Appendix 15A Data Compression Using Zip 475 Appendix 15B Radix-64 Conversion 478 Appendix 15C PGP Random Number Generation 479 Chapter 16. IP Security 483 Section 16.1. IP Security Overview 485 Section 16.2. IP Security Architecture 487 Section 16.3. Authentication Header 493 Section 16.4. Encapsulating Security Payload 498 Section 16.5. Combining Security Associations 503 Section 16.6. Key Management 506 Section 16.7. Recommended Reading and Web Site 516 Section 16.8. Key Terms, Review Questions, and Problems 517 Appendix 16A Internetworking and Internet Protocols 518 Chapter 17. Web Security 527 Section 17.1. Web Security Considerations 528 Section 17.2. Secure Socket Layer and Transport Layer Security 531 Section 17.3. Secure Electronic Transaction 549

Section 17.4. Recommended Reading and Web Sites 560

Section 17.5. Key Terms, Review Questions, and Problems 561

Part Four: System Security 563

Chapter 18. Intruders 565

Section 18.1. Intruders 567

Section 18.2. Intrusion Detection 570

Section 18.3. Password Management 582

Section 18.4. Recommended Reading and Web Sites 591

Section 18.5. Key Terms, Review Questions, and Problems 592

Appendix 18A The Base-Rate Fallacy 594

Chapter 19. Malicious Software 598

Section 19.1. Viruses and Related Threats 599

Section 19.2. Virus Countermeasures 610

Section 19.3. Distributed Denial of Service Attacks 614

Section 19.4. Recommended Reading and Web Sites 619

Section 19.5. Key Terms, Review Questions, and Problems 620

Chapter 20. Firewalls 621

Section 20.1. Firewall Design Principles 622

Section 20.2. Trusted Systems 634

Section 20.3. Common Criteria for Information Technology Security Evaluation640

Section 20.4. Recommended Reading and Web Sites 644

Section 20.5. Key Terms, Review Questions, and Problems 645

Appendix A. Standards and Standards-Setting Organizations 647

Section A.1. The Importance of Standards 648

Section A.2. Internet Standards and the Internet Society 649

Section A.3. National Institute of Standards and Technology 652

Appendix B. Projects for Teaching Cryptography and Network Security 653

Section B.1. Research Projects 654

Section B.2. Programming Projects 655

Section B.3. Laboratory Exercises 655

Section B.4. Writing Assignments 655

Section B.5. Reading/Report Assignments 656

Glossary 657

References 663

Abbreviations 663

Inside Front Cover InsideFrontCover

Inside Back Cover InsideBackCover

Index



**Copyright**

[Page ii]

**Library of Congress Cataloging-in-Publication Data on File**

Vice President and Editorial Director, ECS: *Marcia J. Horton*

Executive Editor: *Tracy Dunkelberger*

Editorial Assistant: *Christianna Lee*

Executive Managing Editor: *Vince O'Brien*

Managing Editor: *Camille Trentacoste*

Production Editor: *Rose Kernan*

Director of Creative Services: *Paul Belfanti*

Cover Designer: *Bruce Kenselaar*

Managing Editor, AV Management and Production: *Patricia Burns*

Art Editor: *Gregory Dulles*

Manufacturing Manager: *Alexis Heydt-Long*

Manufacturing Buyer: *Lisa McDowell*

Marketing Manager: *Robin O'Brien*

Marketing Assistant: *Barrie Reinhold*

© 2006 Pearson Education, Inc.

Pearson Prentice Hall

Pearson Education, Inc.

Upper Saddle River, NJ 07458

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Pearson Prentice Hall™ is a trademark of Pearson Education, Inc.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Pearson Education Ltd., *London*

Pearson Education Australia Pty. Ltd., *Sydney*

Pearson Education Singapore, Pte. Ltd.

Pearson Education North Asia Ltd., *Hong Kong*

Pearson Education Canada, Inc., *Toronto*

Pearson Educacíon de Mexico, S.A. de C.V.

Pearson EducationJapan, *Tokyo*

Pearson Education Malaysia, Pte. Ltd.

Pearson Education Inc., *Upper Saddle River, New Jersey*

[Page iii]

**Dedication**

*To Antigone never dull never boring always a Sage*

**

**[Page xi]

**Notation**

*Even the natives have difficulty mastering this peculiar vocabulary.*

*The Golden Bough*, Sir James George Frazer

| **Symbol** | **Expression** | **Meaning** |
| --- | --- | --- |
| D, *K* | D(*K, Y*) | Symmetric decryption of ciphertext *Y* using secret key *K*. |
| D, *PRa* | D(*PRa*, *Y*) | Asymmetric decryption of ciphertext *Y* using A's private key *PRa* |
| D,*PUa* | D(*PUa*, *Y*) | Asymmetric decryption of ciphertext *Y* using A's public key *PUa* |
| E, *K* | E(*K, X*) | Symmetric encryption of plaintext *X* using secret key *K*. |
| E, *PRa* | E(*PRa*, X) | Asymmetric encryption of plaintext *X* using A's private key *PRa* |
| E, *PUa* | E(*PUa*, X) | Asymmetric encryption of plaintext *X* using A's public key *PUa* |
| *K* |  | Secret key |
| *PRa* |  | Private key of user A |
| *PUa* |  | Public key of user A |
| C, *K* | C(*K, X*) | Message authentication code of message *X* using secret key *K*. |
| GF(*p*) |  | The finite field of order *p*, where *p* is prime. The field is defined as the set *Zp* together with the arithmetic operations modulo *p*. |
| GF(2*n*) |  | The finite field of order 2*n*. |
| *Zn* |  | Set of nonnegative integers less than *n* |
| gcd | gcd(*i*, *j*) | Greatest common divisor; the largest positive integer that divides both *i* and *j* with no remainder on division. |
| mod | *a* mod *m* | Remainder after division of *a* by *m*. |
| mod, | *a b*(mod *m*) | *a* mod *m* = *b* mod *m* |
| mod, | *a b*(mod *m*) | *a* mod *m b* mod *m* |
| dlog | dlog*a,p*(*b*) | Discrete logarithm of the number *b* for the base *a* (mod *p*) |
| φ | φ(*n*) | The number of positive integers less than *n* and relatively prime to *n*. This is Euler's totient function. |

| Σ |  | *a*1 + *a*2 + ... + *a*n |
| --- | --- | --- |
|  |  | *a*1 x *a*2 x ... x *a*n |
| | | *i*|*j* | *i* divides *j*, which means that there is no remainder when *j* is divided by *i* |
| |,| | |*a*| | Absolute value of *a* |
| || | *x*||*y* | *x* concatenated with *y* |
|  | *x y* | *x* is approximately equal to *y* |
|  | *x y* | Exclusive-OR of *x* and *y* for single-bit variables; Bitwise exclusive-OR of *x* and *y* for multiple-bit variables |
| , | *x* | The largest integer less than or equal to *x* |
|  | *x* S | The element *x* is contained in the set S. |
|  | *A* (*a*1,*a*2, ...,*ak*) | The integer A corresponds to the sequence of integers (*a*1,*a*2, ...,*ak*) |



[Page xiii]

**Preface**

*"The tie, if I might suggest it, sir, a shade more tightly knotted. One aims at the perfect butterfly effect. If you will permit me"*

*"What does it matter, Jeeves, at a time like this? Do you realize that Mr. Little's domestic happiness is hanging in the scale?"*

*"There is no time, sir, at which ties do not matter."*

*Very Good, Jeeves!* P. G. Wodehouse

In this age of universal electronic connectivity, of viruses and hackers, of electronic eavesdropping and electronic fraud, there is indeed no time at which security does not matter. Two trends have come together to make the topic of this book of vital interest. First, the explosive growth in computer systems and their interconnections via networks has increased the dependence of both organizations and individuals on the information stored and communicated using these systems. This, in turn, has led to a heightened awareness of the need to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks. Second, the disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications to enforce network security.



[Page xiii (continued)]

**Objectives**

It is the purpose of this book to provide a practical survey of both the principles and practice of cryptography and network security. In the first two parts of the book, the basic issues to be addressed by a network security capability are explored by providing a tutorial and survey of cryptography and network security technology. The latter part of the book deals with the practice of network security: practical applications that have been implemented and are in use to provide network security.

The subject, and therefore this book, draws on a variety of disciplines. In particular, it is impossible to appreciate the significance of some of the techniques discussed in this book without a basic understanding of number theory and some results from probability theory. Nevertheless, an attempt has been made to make the book self-contained. The book presents not only the basic mathematical results that are needed but provides the reader with an intuitive understanding of those results. Such background material is introduced as needed. This approach helps to motivate the material that is introduced, and the author considers this preferable to simply presenting all of the mathematical material in a lump at the beginning of the book.





[Page xiii (continued)]

**Intended Audience**

The book is intended for both an academic and a professional audience. As a textbook, it is intended as a one-semester undergraduate course in cryptography and network security for computer science, computer engineering, and electrical engineering majors. It covers the material in IAS2 Security Mechanisms, a core area in the Information Technology body of knowledge; NET4 Security, another core area in the Information Technology body of knowledge; and IT311, Cryptography, an advanced course; these subject areas are part of the Draft ACM/IEEE Computer Society Computing Curricula 2005.

[Page xiv]

The book also serves as a basic reference volume and is suitable for self-study.





[Page xiv (continued)]

**Plan of the Book**

The book is organized in four parts:

**Part One. Conventional Encryption:** A detailed examination of conventional encryption algorithms and design principles, including a discussion of the use of conventional encryption for confidentiality.

**Part Two. Public-Key Encryption and Hash Functions:** A detailed examination of public-key encryption algorithms and design principles. This part also examines the use of message authentication codes and hash functions, as well as digital signatures and public key certificates.

**Part Three. Network Security Practice:** Covers important network security tools and applications, including Kerberos, X.509v3 certificates, PGP, S/MIME, IP Security, SSL/TLS, and SET.

**Part Four. System Security:** Looks at system-level security issues, including the threat of and countermeasures for intruders and viruses, and the use of firewalls and trusted systems.

In addition, the book includes an extensive glossary, a list of frequently used acronyms, and a bibliography. Each chapter includes homework problems, review questions, a list of key words, suggestions for further reading, and recommended Web sites.

A more detailed, chapter-by-chapter summary of each part appears at the beginning of that part.

[Page xiv (continued)]

**Internet Services for Instructors and Students**

There is a Web site for this book that provides support for students and instructors. The site includes links to other relevant sites, transparency masters of figures and tables in the book in PDF (Adobe Acrobat) format, and PowerPoint slides. The Web page is at WilliamStallings.com/Crypto/Crypto4e.html. As soon as typos or other errors are discovered, an errata list for this book will be available at WilliamStallings.com. In addition, the Computer Science Student Resource site, at WilliamStallings.com/ StudentSupport.html, provides documents, information, and useful links for computer science students and professionals.



[Page xiv (continued)]

**Projects for Teaching Cryptography and Network Security**

For many instructors, an important component of a cryptography or security course is a project or set of projects by which the student gets hands-on experience to reinforce concepts from the text. This book provides an unparalleled degree of support for including a projects component in the course. The instructor's manual not only includes guidance on how to assign and structure the projects, but also includes a set of suggested projects that covers a broad range of topics from the text:

[Page xv]

● **Research projects:** A series of research assignments that instruct the student to research a particular topic on the Internet and write a report

● **Programming projects:** A series of programming projects that cover a broad range of topics and that can be implemented in any suitable language on any platform

● **Lab exercises:** A series of projects that involve programming and experimenting with concepts from the book

● **Writing assignments:** A set of suggested writing assignments, by chapter

● **Reading/report assignments:** A list of papers in the literature, one for each chapter, that can be assigned for the student to read and then write a short report

See Appendix B for details.



[Page xv (continued)]

**What's New in the Fourth Edition**

In the three years since the third edition of this book was published, the field has seen continued innovations and improvements. In this new edition, I try to capture these changes while maintaining a broad and comprehensive coverage of the entire field. To begin this process of revision, the third edition was extensively reviewed by a number of professors who teach the subject. In addition, a number of professionals working in the field reviewed individual chapters. The result is that, in many places, the narrative has been clarified and tightened, and illustrations have been improved. Also, a large number of new "field-tested" problems have been added.

Beyond these refinements to improve pedagogy and user friendliness, there have been major substantive changes throughout the book. Highlights include the following:

● **Simplified AES:** This is an educational, simplified version of AES (Advanced Encryption Standard), which enables students to grasp the essentials of AES more easily.

● **Whirlpool:** This is an important new secure hash algorithm based on the use of a symmetric block cipher.

● **CMAC:** This is a new block cipher mode of operation. CMAC (cipher-based message authentication code) provides message authentication based on the use of a symmetric block cipher.

● **Public-key infrastructure (PKI):** This important topic is treated in this new edition. ● **Distributed denial of service (DDoS) attacks:** DDoS attacks have assumed increasing significance in recent years.

● **Common Criteria for Information Technology Security Evaluation:** The Common Criteria have become the international framework for expressing security requirements and evaluating products and implementations.

● **Online appendices:** Six appendices available at this book's Web site supplement the material in the text.

In addition, much of the other material in the book has been updated and revised.

[Page xvi]

**Acknowledgments**

This new edition has benefited from review by a number of people, who gave generously of their time and expertise. The following people reviewed all or a large part of the manuscript: Danny Krizanc (Wesleyan University), Breno de Medeiros (Florida State University), Roger H. Brown (Rensselaer at Hartford), Cristina Nita-Rotarul (Purdue University), and Jimmy McGibney (Waterford Institute of Technology).

Thanks also to the many people who provided detailed technical reviews of a single chapter: Richard Outerbridge, Jorge Nakahara, Jeroen van de Graaf, Philip Moseley, Andre Correa, Brian Bowling, James Muir, Andrew Holt, Décio Luiz Gazzoni Filho, Lucas Ferreira, Dr. Kemal Bicakci, Routo Terada, Anton Stiglic, Valery Pryamikov, and Yongge Wang.

Joan Daemen kindly reviewed the chapter on AES. Vincent Rijmen reviewed the material on Whirlpool. And Edward F. Schaefer reviewed the material on simplified AES.

The following people contributed homework problems for the new edition: Joshua Brandon Holden (Rose Hulman Institute if Technology), Kris Gaj (George Mason University), and James Muir (University of Waterloo).

Sanjay Rao and Ruben Torres of Purdue developed the laboratory exercises that appear in the instructor's supplement. The following people contributed project assignments that appear in the instructor's supplement: Henning Schulzrinne (Columbia University); Cetin Kaya Koc (Oregon State University); and David Balenson (Trusted Information Systems and George Washington University).

Finally, I would like to thank the many people responsible for the publication of the book, all of whom did their usual excellent job. This includes the staff at Prentice Hall, particularly production manager Rose Kernan; my supplements manager Sarah Parker; and my new editor Tracy Dunkelberger. Also, Patricia M. Daly did the copy editing.

With all this assistance, little remains for which I can take full credit. However, I am proud to say that, with no help whatsoever, I selected all of the quotations.



[Page 1]

**Chapter 0. Reader's Guide**

**0.1 Outline of this Book**

**0.2 Roadmap**

Subject Matter

Topic Ordering

**0.3 Internet and Web Resources**

Web Sites for This Book

Other Web Sites

USENET Newsgroups

[Page 2]

*The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable.*

*The Art of War*, Sun Tzu

This book, with its accompanying Web site, covers a lot of material. Here we give the reader an overview.



[Page 2 (continued)]

**0.1. Outline of this Book**

Following an introductory chapter, Chapter 1, the book is organized into four parts:

**Part One: Symmetric Ciphers:** Provides a survey of symmetric encryption, including classical and modern algorithms. The emphasis is on the two most important algorithms, the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). This part also addresses message authentication and key management.

**Part Two: Public-Key Encryption and Hash Functions:** Provides a survey of public key algorithms, including RSA (Rivest-Shamir-Adelman) and elliptic curve. It also covers public-key applications, including digital signatures and key exchange.

**Part Three: Network Security Practice:** Examines the use of cryptographic algorithms and security protocols to provide security over networks and the Internet. Topics covered include user authentication, e-mail, IP security, and Web security.

**Part Four: System Security:** Deals with security facilities designed to protect a computer system from security threats, including intruders, viruses, and worms. This part also looks at firewall technology.

Many of the cryptographic algorithms and network security protocols and applications described in this book have been specified as standards. The most important of these are Internet Standards, defined in Internet RFCs (Request for Comments), and Federal Information Processing Standards (FIPS), issued by the National Institute of Standards and Technology (NIST). Appendix A discusses the standards-making process and lists the standards cited in this book.



[Page 2 (continued)]

**0.2. Roadmap**

**Subject Matter**

The material in this book is organized into three broad categories:

**Cryptology:** This is the study of techniques for ensuring the secrecy and/or authenticity of information. The two main branches of cryptology are **cryptography**, which is the study of the design of such techniques; and **cryptanalysis**, which deals with the defeating such techniques, to recover information, or forging information that will be accepted as authentic.

[Page 3]

**Network security:** This area covers the use of cryptographic algorithms in network protocols and network applications.

**Computer security:** In this book, we use this term to refer to the security of computers against intruders (e.g., hackers) and malicious software (e.g., viruses). Typically, the computer to be secured is attached to a network and the bulk of the threats arise from the network.

The first two parts of the book deal with two distinct cryptographic approaches: symmetric cryptographic algorithms and public-key, or asymmetric, cryptographic algorithms. Symmetric algorithms make use of a single shared key shared by two parties. Public-key algorithms make use of two keys: a private key known only to one party, and a public key, available to other parties.

**Topic Ordering**

This book covers a lot of material. For the instructor or reader who wishes a shorter treatment, there are a number of opportunities.

To thoroughly cover the material in the first two parts, the chapters should be read in sequence. With the exception of the Advanced Encryption Standard (AES), none of the material in **Part One** requires any special mathematical background. To understand AES, it is necessary to have some understanding of finite fields. In turn, an understanding of finite fields requires a basic background in prime numbers and modular arithmetic. Accordingly, Chapter 4 covers all of these mathematical preliminaries just prior to their use in Chapter 5 on AES. Thus, if Chapter 5 is skipped, it is safe to skip Chapter 4 as well.

Chapter 2 introduces some concepts that are useful in later chapters of Part One. However, for the reader whose sole interest is contemporary cryptography, this chapter can be quickly skimmed. The two most important symmetric cryptographic algorithms are DES and AES, which are covered in Chapters 3 and 5, respectively. Chapter 6 covers two other interesting algorithms, both of which enjoy commercial use. This chapter can be safely skipped if these algorithms are not of interest.

For **Part Two**, the only additional mathematical background that is needed is in the area of number

theory, which is covered in Chapter 8. The reader who has skipped Chapters 4 and 5 should first review the material on Sections 4.1 through 4.3.

The two most widely used general-purpose public-key algorithms are RSA and elliptic curve, with RSA enjoying much wider acceptance. The reader may wish to skip the material on elliptic curve cryptography in Chapter 10, at least on a first reading. In Chapter 12, Whirlpool and CMAC are of lesser importance.

**Part Three** and **Part Four** are relatively independent of each other and can be read in either order. Both parts assume a basic understanding of the material in Parts One and Two.





[Page 4]

**0.3. Internet and Web Resources**

There are a number of resources available on the Internet and the Web to support this book and to help one keep up with developments in this field.

**Web Sites for This Book**

A special Web page has been set up for this book at **WilliamStallings.com/Crypto/Crypto4e.html.** The site includes the following:

● **Useful Web sites:** There are links to other relevant Web sites, organized by chapter, including the sites listed in this section and throughout this book.

● **Errata sheet:** An errata list for this book will be maintained and updated as needed. Please e mail any errors that you spot to me. Errata sheets for my other books are at **WilliamStallings. com.**

● **Figures:** All of the figures in this book in PDF (Adobe Acrobat) format.

● **Tables:** All of the tables in this book in PDF format.

● **Slides:** A set of PowerPoint slides, organized by chapter.

● **Cryptography and network security courses:** There are links to home pages for courses based on this book; these pages may be useful to other instructors in providing ideas about how to structure their course.

I also maintain the Computer Science Student Resource Site, at **WilliamStallings.com/ StudentSupport.html.** The purpose of this site is to provide documents, information, and links for computer science students and professionals. Links and documents are organized into four categories:

● **Math:** Includes a basic math refresher, a queuing analysis primer, a number system primer, and links to numerous math sites

● **How-to:** Advice and guidance for solving homework problems, writing technical reports, and preparing technical presentations

● **Research resources:** Links to important collections of papers, technical reports, and bibliographies

● **Miscellaneous:** A variety of other useful documents and links

**Other Web Sites**

There are numerous Web sites that provide information related to the topics of this book. In subsequent chapters, pointers to specific Web sites can be found in the *Recommended Reading and Web Sites* section. Because the addresses for Web sites tend to change frequently, I have not included URLs in the book. For all of the Web sites listed in the book, the appropriate link can be found at this book's Web site. Other links not mentioned in this book will be added to the Web site over time.

[Page 5]

**USENET Newsgroups**

A number of USENET newsgroups are devoted to some aspect of cryptography or network security. As

with virtually all USENET groups, there is a high noise-to-signal ratio, but it is worth experimenting to see if any meet your needs. The most relevant are

● **sci.crypt.research:** The best group to follow. This is a moderated newsgroup that deals with research topics; postings must have some relationship to the technical aspects of cryptology. ● **sci.crypt:** A general discussion of cryptology and related topics.

● **sci.crypt.random-numbers:** A discussion of cryptographic-strength random number generators. ● **alt.security:** A general discussion of security topics.

● **comp.security.misc:** A general discussion of computer security topics.

● **comp.security.firewalls:** A discussion of firewall products and technology.

● **comp.security.announce:** News, announcements from CERT.

● **comp.risks:** A discussion of risks to the public from computers and users.

● **comp.virus:** A moderated discussion of computer viruses.



[Page 6]

**Chapter 1. Introduction**

**1.1 Security Trends**

**1.2 The OSI Security Architecture**

**1.3 Security Attacks**

Passive Attacks

Active Attacks

**1.4 Security Services**

Authentication

Access Control

Data Confidentiality

Data Integrity

Nonrepudiation

Availability Service

**1.5 Security Mechanisms**

**1.6 A Model for Network Security**

**1.7 Recommended Reading and Web Sites**

**1.8 Key Terms, Review Questions, and Problems**

Key Terms

Review Questions

Problems

[Page 7]

*The combination of space, time, and strength that must be considered as the basic elements of this theory of defense makes this a fairly complicated matter. Consequently, it is not easy to find a fixed point of departure*.

*On War*, Carl Von Clausewitz

| **Key Points**  ● The OSI (open systems interconnection) security architecture provides a systematic framework for defining security attacks, mechanisms, and services.  ● **Security attacks** are classified as either passive attacks, which include unauthorized reading of a message of file and traffic analysis; and active attacks, such as modification of messages or files, and denial of service.  ● A **security mechanism** is any process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack. Examples of mechanisms are encryption algorithms, digital signatures, and authentication protocols.  ● **Security services** include authentication, access control, data confidentiality, data integrity, nonrepudiation, and availability. |
| --- |

The requirements of **information security** within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. An example of the former is the use of rugged filing cabinets with a combination lock for storing sensitive documents. An example of the latter is personnel screening procedures used during the hiring process.

With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This is especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone network, data network, or the Internet. The generic name for the collection of tools designed to protect data and to thwart hackers is **computer security**.

The second major change that affected security is the introduction of distributed systems and the use of networks and communications facilities for carrying data between terminal user and computer and between computer and computer. Network security measures are needed to protect data during their transmission. In fact, the term **network security** is somewhat misleading, because virtually all business, government, and academic organizations interconnect their data processing equipment with a collection of interconnected networks. Such a collection is often referred to as an internet,[1]and the term **internet security** is used.

[1] We use the term *internet*, with a lowercase "i," to refer to any interconnected collection of networks. A corporate intranet is an example of an internet. The Internet with a capital "I" may be one of the facilities used by an organization to construct its internet.

[Page 8]

There are no clear boundaries between these two forms of security. For example, one of the most publicized types of attack on information systems is the computer virus. A virus may be introduced into a system physically when it arrives on a diskette or optical disk and is subsequently loaded onto a computer. Viruses may also arrive over an internet. In either case, once the virus is resident on a computer system, internal computer security tools are needed to detect and recover from the virus.

This book focuses on internet security, which consists of measures to deter, prevent, detect, and correct security violations that involve the transmission of information. That is a broad statement that covers a host of possibilities. To give you a feel for the areas covered in this book, consider the following examples of security violations:

**1.**

User A transmits a file to user B. The file contains sensitive information (e.g., payroll records) that is to be protected from disclosure. User C, who is not authorized to read the file, is able to monitor the transmission and capture a copy of the file during its transmission.

**2.**

A network manager, D, transmits a message to a computer, E, under its management. The message instructs computer E to update an authorization file to include the identities of a number of new users who are to be given access to that computer. User F intercepts the message, alters its contents to add or delete entries, and then forwards the message to E, which accepts the message as coming from manager D and updates its authorization file accordingly.

**3.**

Rather than intercept a message, user F constructs its own message with the desired entries and transmits that message to E as if it had come from manager D. Computer E accepts the message as coming from manager D and updates its authorization file accordingly.

**4.**

An employee is fired without warning. The personnel manager sends a message to a server system to invalidate the employee's account. When the invalidation is accomplished, the server is to post a notice to the employee's file as confirmation of the action. The employee is able to intercept the message and delay it long enough to make a final access to the server to retrieve sensitive information. The message is then forwarded, the action taken, and the confirmation posted. The employee's action may go unnoticed for some considerable time.

**5.**

A message is sent from a customer to a stockbroker with instructions for various transactions. Subsequently, the investments lose value and the customer denies sending the message.

Although this list by no means exhausts the possible types of security violations, it illustrates the range of concerns of network security.

[Page 9]

Internetwork security is both fascinating and complex. Some of the reasons follow:

Security involving communications and networks is not as simple as it might first appear to the novice. The requirements seem to be straightforward; indeed, most of the major requirements for security services can be given self-explanatory one-word labels: confidentiality, authentication, nonrepudiation, integrity. But the mechanisms used to meet those requirements can be quite complex, and understanding them may involve rather subtle reasoning.

**2.**

In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features. In many cases, successful attacks are designed by looking at the problem in a completely different way, therefore exploiting an unexpected weakness in the mechanism.

**3.**

Because of point 2, the procedures used to provide particular services are often counterintuitive: It is not obvious from the statement of a particular requirement that such elaborate measures are needed. It is only when the various countermeasures are considered that the measures used make sense.

**4.**

Having designed various security mechanisms, it is necessary to decide where to use them. This is true both in terms of physical placement (e.g., at what points in a network are certain security mechanisms needed) and in a logical sense [e.g., at what layer or layers of an architecture such as TCP/IP (Transmission Control Protocol/Internet Protocol) should mechanisms be placed].

**5.**

Security mechanisms usually involve more than a particular algorithm or protocol. They usually also require that participants be in possession of some secret information (e.g., an encryption key), which raises questions about the creation, distribution, and protection of that secret information. There is also a reliance on communications protocols whose behavior may complicate the task of developing the security mechanism. For example, if the proper functioning of the security mechanism requires setting time limits on the transit time of a message from sender to receiver, then any protocol or network that introduces variable, unpredictable delays may render such time limits meaningless.

Thus, there is much to consider. This chapter provides a general overview of the subject matter that structures the material in the remainder of the book. We begin with a general discussion of network security services and mechanisms and of the types of attacks they are designed for. Then we develop a general overall model within which the security services and mechanisms can be viewed.







[Page 9 (continued)]

**1.1. Security Trends**

In 1994, the Internet Architecture Board (IAB) issued a report entitled "Security in the Internet Architecture" (RFC 1636). The report stated the general consensus that the Internet needs more and better security, and it identified key areas for security mechanisms. Among these were the need to secure the network infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

[Page 10]

These concerns are fully justified. As confirmation, consider the trends reported by the Computer Emergency Response Team (CERT) Coordination Center (CERT/CC). Figure 1.1a shows the trend in Internet-related vulnerabilities reported to CERT over a 10-year period. These include security weaknesses in the operating systems of attached computers (e.g., Windows, Linux) as well as vulnerabilities in Internet routers and other network devices. Figure 1.1b shows the number of security related incidents reported to CERT. These include denial of service attacks; IP spoofing, in which intruders create packets with false IP addresses and exploit applications that use authentication based on IP; and various forms of eavesdropping and packet sniffing, in which attackers read transmitted information, including logon information and database contents.

[Page 11]

**Figure 1.1. CERT Statistics**

**(This item is displayed on page 10 in the print version)**

[View full size image]



Over time, the attacks on the Internet and Internet-attached systems have grown more sophisticated while the amount of skill and knowledge required to mount an attack has declined (Figure 1.2). Attacks have become more automated and can cause greater amounts of damage.

**Figure 1.2. Trends in Attack Sophistication and Intruder Knowledge**

[View full size image]



This increase in attacks coincides with an increased use of the Internet and with increases in the complexity of protocols, applications, and the Internet itself. Critical infrastructures increasingly rely on the Internet for operations. Individual users rely on the security of the Internet, email, the Web, and Web-based applications to a greater extent than ever. Thus, a wide range of technologies and tools are needed to counter the growing threat. At a basic level, cryptographic algorithms for confidentiality and authentication assume greater importance. As well, designers need to focus on Internet-based protocols and the vulnerabilities of attached operating systems and applications. This book surveys all of these technical areas.





[Page 12]

**1.2. The OSI Security Architecture**

To assess effectively the security needs of an organization and to evaluate and choose various security products and policies, the manager responsible for security needs some systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. This is difficult enough in a centralized data processing environment; with the use of local and wide area networks, the problems are compounded.

ITU-T[2]Recommendation X.800, *Security Architecture for OSI*, defines such a systematic approach.[3] The OSI security architecture is useful to managers as a way of organizing the task of providing security. Furthermore, because this architecture was developed as an international standard, computer and communications vendors have developed security features for their products and services that relate to this structured definition of services and mechanisms.

[2] The International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) is a United Nationssponsored agency that develops standards, called Recommendations, relating to telecommunications and to open systems interconnection (OSI).

[3] The OSI security architecture was developed in the context of the OSI protocol architecture, which is described in Appendix H. However, for our purposes in this chapter, an understanding of the OSI protocol architecture is not required.

For our purposes, the OSI security architecture provides a useful, if abstract, overview of many of the concepts that this book deals with. The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as follows:

● **Security attack:** Any action that compromises the security of information owned by an organization.

● **Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

● **Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

In the literature, the terms *threat* and *attack* are commonly used to mean more or less the same thing. Table 1.1 provides definitions taken from RFC 2828, *Internet Security Glossary*.

**Table 1.1. Threats and Attacks (RFC 2828)**

| **Threat** |
| --- |
| A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability. |
| **Attack** |

| An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system. |
| --- |



[Page 13]

**1.3. Security Attacks**

A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks*. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

**Passive Attacks**

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are release of message contents and traffic analysis.

The **release of message contents** is easily understood (Figure 1.3a). A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

**Figure 1.3. Passive Attacks**

**(This item is displayed on page 14 in the print version)**

[View full size image]



A second type of passive attack, **traffic analysis**, is subtler (Figure 1.3b). Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus,

the emphasis in dealing with passive attacks is on prevention rather than detection. **Active Attacks**

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

A **masquerade** takes place when one entity pretends to be a different entity (Figure 1.4a). A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

**Figure 1.4. Active Attacks**

**(This item is displayed on pages 15 - 16 in the print version)**

[View full size image]

**Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (Figure 1.4b).

[Page 14]

**Modification of messages** simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure 1.4c). For example, a message meaning "Allow John Smith to read confidential file *accounts*" is modified to mean "Allow Fred Brown to read confidential file *accounts.*"

The **denial of service** prevents or inhibits the normal use or management of communications facilities (Figure 1.4d). This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

[Page 15]

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely, because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention.



[Page 16]

**1.4. Security Services**

X.800 defines a security service as a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers. Perhaps a clearer definition is found in RFC 2828, which provides the following definition: a processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms.

[Page 17]

X.800 divides these services into five categories and fourteen specific services (Table 1.2). We look at each category in turn.[4]

[4] There is no universal agreement about many of the terms used in the security literature. For example, the term *integrity* is sometimes used to refer to all aspects of information security. The term *authentication* is sometimes used to refer both to verification of identity and to the various functions listed under integrity in this chapter. Our usage here agrees with both X.800 and RFC 2828.

**Table 1.2. Security Services (X.800)**

| **AUTHENTICATION** |
| --- |
| The assurance that the communicating entity is the one that it claims to be. |
| **Peer Entity Authentication** |
| Used in association with a logical connection to provide confidence in the identity of the entities connected. |
| **Data Origin Authentication** |
| In a connectionless transfer, provides assurance that the source of received data is as claimed. |
| **ACCESS CONTROL** |
| The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do). |
| **DATA CONFIDENTIALITY** |
| The protection of data from unauthorized disclosure. |

| **Connection Confidentiality** |
| --- |
| The protection of all user data on a connection. |
| **Connectionless Confidentiality** |
| The protection of all user data in a single data block |
| **Selective-Field Confidentiality** |
| The confidentiality of selected fields within the user data on a connection or in a single data block. |
| **Traffic Flow Confidentiality** |
| The protection of the information that might be derived from observation of traffic flows. |
| **DATA INTEGRITY** |
| The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay). |
| **Connection Integrity with Recovery** |
| Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted. |
| **Connection Integrity without Recovery** |
| As above, but provides only detection without recovery. |
| **Selective-Field Connection Integrity** |
| Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed. |
| **Connectionless Integrity** |
| Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided. |
| **Selective-Field Connectionless Integrity** |
| Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified. |
| **NONREPUDIATION** |
| Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication. |
| **Nonrepudiation, Origin** |
| Proof that the message was sent by the specified party. |
| **Nonrepudiation, Destination** |
| Proof that the message was received by the specified party. |

[Page 18]

**Authentication**

The authentication service is concerned with assuring that a communication is authentic. In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from. In the case of an ongoing interaction, such as the connection of a terminal to a host, two aspects are involved. First, at the time of connection initiation, the service assures that the two entities are authentic, that is, that each is the entity that it claims to be. Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unauthorized transmission or reception.

Two specific authentication services are defined in X.800:

● **Peer entity authentication:** Provides for the corroboration of the identity of a peer entity in an association. It is provided for use at the establishment of, or at times during the data transfer phase of, a connection. It attempts to provide confidence that an entity is not performing either a masquerade or an unauthorized replay of a previous connection.

● **Data origin authentication:** Provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units. This type of service supports applications like electronic mail where there are no prior interactions between the communicating entities.

**Access Control**

In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

**Data Confidentiality**

Confidentiality is the protection of transmitted data from passive attacks. With respect to the content of a data transmission, several levels of protection can be identified. The broadest service protects all user data transmitted between two users over a period of time. For example, when a TCP connection is set up between two systems, this broad protection prevents the release of any user data transmitted over the TCP connection. Narrower forms of this service can also be defined, including the protection of a single message or even specific fields within a message. These refinements are less useful than the broad approach and may even be more complex and expensive to implement.

The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility.

**Data Integrity**

As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. Again, the most useful and straightforward approach is total stream protection.

[Page 19]

A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent, with no duplication, insertion, modification, reordering, or replays. The destruction of data is also covered under this service. Thus, the connection-oriented integrity service addresses both message stream modification and denial of service. On the other hand, a connectionless integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only.

We can make a distinction between the service with and without recovery. Because the integrity service relates to active attacks, we are concerned with detection rather than prevention. If a violation of integrity is detected, then the service may simply report this violation, and some other portion of software or human intervention is required to recover from the violation. Alternatively, there are mechanisms available to recover from the loss of integrity of data, as we will review subsequently. The incorporation of automated recovery mechanisms is, in general, the more attractive alternative.

**Nonrepudiation**

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

**Availability Service**

Both X.800 and RFC 2828 define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system (i.e., a system is available if it provides services according to the system design whenever users request them). A variety of attacks can result in the loss of or reduction in availability. Some of these attacks are amenable to automated countermeasures, such as authentication and encryption, whereas others require some sort of physical action to prevent or recover from loss of availability of elements of a distributed system.

X.800 treats availability as a property to be associated with various security services. However, it makes sense to call out specifically an availability service. An availability service is one that protects a system to ensure its availability. This service addresses the security concerns raised by denial-of-service attacks. It depends on proper management and control of system resources and thus depends on access control service and other security services.



[Page 19 (continued)]

**1.5. Security Mechanisms**

Table 1.3 lists the security mechanisms defined in X.800. As can be seen the mechanisms are divided into those that are implemented in a specific protocol layer and those that are not specific to any particular protocol layer or security service. These mechanisms will be covered in the appropriate places in the book and so we do not elaborate now, except to comment on the definition of encipherment. X.800 distinguishes between reversible encipherment mechanisms and irreversible encipherment mechanisms. A reversible encipherment mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted. Irreversible encipherment mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications.

[Page 20]

**Table 1.3. Security Mechanisms (X.800)**

| **SPECIFIC SECURITY MECHANISMS** |
| --- |
| May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services. |
| **Encipherment** |
| The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys. |
| **Digital Signature** |
| Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient). |
| **Access Control** |
| A variety of mechanisms that enforce access rights to resources. |
| **Data Integrity** |
| A variety of mechanisms used to assure the integrity of a data unit or stream of data units. |
| **Authentication Exchange** |
| A mechanism intended to ensure the identity of an entity by means of information exchange. |
| **Traffic Padding** |
| The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts. |
| **Routing Control** |
| Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected. |
| **Notarization** |
| The use of a trusted third party to assure certain properties of a data exchange. |
| **PERVASIVE SECURITY MECHANISMS** |
| Mechanisms that are not specific to any particular OSI security service or protocol layer. |
| **Trusted Functionality** |
| That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy). |

| **Security Label** |
| --- |
| The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource. |
| **Event Detection** |
| Detection of security-relevant events. |
| **Security Audit Trail** |
| Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities. |
| **Security Recovery** |
| Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions. |

Table 1.4, based on one in X.800, indicates the relationship between security services and security mechanisms. [Page 21]

**Table 1.4. Relationship between Security Services and Mechanisms**

| **Mechanism** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Service** | **Encipherment** | **Digital**  **Signature** | **Access Control** | **Data**  **Integrity** | **Authentication Exchange** | **Traffic**  **Padding** | **Routing Control** | **Notarization** |
| Peer entity  authentication | Y | Y |  |  | Y |  |  |  |
| Data origin  authentication | Y | Y |  |  |  |  |  |  |
| Access control |  |  | Y |  |  |  |  |  |
| Confidentiality | Y |  |  |  |  |  | Y |  |
| Traffic flow  confidentiality | Y |  |  |  |  | Y | Y |  |
| Data integrity | Y | Y |  | Y |  |  |  |  |
| Nonrepudiation |  | Y |  | Y |  |  |  | Y |
| Availability |  |  |  | Y | Y |  |  |  |







[Page 22]

**1.6. A Model for Network Security**

A model for much of what we will be discussing is captured, in very general terms, in Figure 1.5. A message is to be transferred from one party to another across some sort of internet. The two parties, who are the *principals* in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

**Figure 1.5. Model for Network Security**

[View full size image]



Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

● A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender

● Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.[5]

[5] Part Two discusses a form of encryption, known as public-key encryption, in which only one of the two principals needs to have the secret information.

[Page 23]

A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent. Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

This general model shows that there are four basic tasks in designing a particular security service: **1.**

Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.

**2.**

Generate the secret information to be used with the algorithm.

**3.**

Develop methods for the distribution and sharing of the secret information.

**4.**

Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

Parts One through Three of this book concentrates on the types of security mechanisms and services that fit into the model shown in Figure 1.5. However, there are other security-related situations of interest that do not neatly fit this model but that are considered in this book. A general model of these other situations is illustrated by Figure 1.6, which reflects a concern for protecting an information system from unwanted access. Most readers are familiar with the concerns caused by the existence of hackers, who attempt to penetrate systems that can be accessed over a network. The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system. Or, the intruder can be a disgruntled employee who wishes to do damage, or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).

**Figure 1.6. Network Access Security Model**

[View full size image]



Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and compilers. Programs can present two kinds of threats:

● **Information access threats** intercept or modify data on behalf of users who should not have access to that data.

● **Service threats** exploit service flaws in computers to inhibit use by legitimate users.

[Page 24]

Viruses and worms are two examples of software attacks. Such attacks can be introduced into a system by means of a disk that contains the unwanted logic concealed in otherwise useful software. They can also be inserted into a system across a network; this latter mechanism is of more concern in network security.

The security mechanisms needed to cope with unwanted access fall into two broad categories (see Figure 1.6). The first category might be termed a gatekeeper function. It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses, and other similar attacks. Once either an unwanted user or unwanted software gains access, the second line of defense consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders. These issues are explored in Part Four.





[Page 24 (continued)]

**1.7. Recommended Reading and Web Sites**

[PFLE02] provides a good introduction to both computer and network security. Two other excellent surveys are [PIEP03] and [BISH05]. [BISH03] covers much the same ground as [BISH05] but with more mathematical detail and rigor. [SCHN00] is valuable reading for any practitioner in the field of computer or network security: it discusses the limitations of technology, and cryptography in particular, in providing security, and the need to consider the hardware, the software implementation, the networks, and the people involved in providing and attacking security.

| **BISH03** Bishop, M. *Computer Security: Art and Science*. Boston: Addison-Wesley, 2003.  **BISH05** Bishop, M. *Introduction to Computer Security*. Boston: Addison-Wesley, 2005.  **PFLE02** Pfleeger, C. *Security in Computing*. Upper Saddle River, NJ: Prentice Hall, 2002.  **PIEP03** Pieprzyk, J.; Hardjono, T.; and Seberry, J. *Fundamentals of Computer Security*. New York: Springer-Verlag, 2003.  **SCHN00** Schneier, B. *Secrets and Lies: Digital Security in a Networked World*. New York: Wiley 2000. |
| --- |

**Recommended Web Sites**

The following Web sites[6]are of general interest related to cryptography and network security:

[6] Because URLs sometimes change, they are not included. For all of the Web sites listed in this and subsequent chapters, the appropriate link is at this book's Web site at **williamstallings.com/Crypto/Crypto4e.html**.

● **COAST:** Comprehensive set of links related to cryptography and network security. ● **IETF Security Area:** Material related to Internet security standardization efforts. ● **Computer and Network Security Reference Index:** A good index to vendor and commercial products, FAQs, newsgroup archives, papers, and other Web sites.

[Page 25]

● **The Cryptography FAQ:** Lengthy and worthwhile FAQ covering all aspects of cryptography. ● **Tom Dunigan's Security Page:** An excellent list of pointers to cryptography and network security Web sites.

● **Helgar Lipma's Cryptology Pointers:** Another excellent list of pointers to cryptography and network security Web sites.

● **IEEE Technical Committee on Security and Privacy:** Copies of their newsletter, information on IEEE-related activities.

● **Computer Security Resource Center:** Maintained by the National Institute of Standards and Technology (NIST); contains a broad range of information on security threats, technology, and standards.

● **Security Focus:** A wide variety of security information, with an emphasis on vendor products and end-user concerns.

● **SANS Institute:** Similar to Security Focus. Extensive collection of white papers.

[Page 25 (continued)]

**1.8. Key Terms, Review Questions, and Problems**

**Key Terms**

access control

active threat

authentication

authenticity

availability

data confidentiality

data integrity

denial of service

encryption

integrity

intruder

masquerade

nonrepudiation

OSI security architecture

passive threat

replay

security attacks

security mechanisms

security services

traffic analysis

**Review Questions**

**1.1** What is the OSI security architecture?

**1.2** What is the difference between passive and active security threats?

**1.3** List and briefly define categories of passive and active security attacks.

**1.4** List and briefly define categories of security services.

**1.5** List and briefly define categories of security mechanisms.

**Problems**

**1.1** Draw a matrix similar to Table 1.4 that shows the relationship between security services and attacks.

**1.2** Draw a matrix similar to Table 1.4 that shows the relationship between security mechanisms and attacks.





[Page 26]

**Part One: Symmetric Ciphers**

*Cryptography is probably the most important aspect of communications*

*security and is becoming increasingly important as a basic building block*

*for computer security.*

*Computers at Risk: Safe Computing in the Information Age*, National

Research Council, 1991

*The increased use of computer and communications systems by industry*

*has increased the risk of theft of proprietary information. Although these*

*threats may require a variety of countermeasures, encryption is a primary*

*method of protecting valuable electronic information.*

*Communications Privacy: Federal Policy and Actions*, General Accounting

Office Report GAO/OSI-94-2, November 1993

By far the most important automated tool for network and communications security is encryption. Two forms of encryption are in common use: conventional, or symmetric, encryption and public-key, or asymmetric, encryption. Part One provides a survey of the basic principles of symmetric encryption, looks at widely used algorithms, and discusses applications of symmetric cryptography.

**Road Map for Part One**

**Chapter 2: Classical Encryption Techniques**

Chapter 2 describes classical symmetric encryption techniques. It provides a

gentle and interesting introduction to cryptography and cryptanalysis and

highlights important concepts.

[Page 27]

**Chapter 3: Block Ciphers and the Data Encryption Standard**

Chapter 3 introduces the principles of modern symmetric cryptography, with

an emphasis on the most widely used encryption technique, the Data

Encryption Standard (DES). The chapter includes a discussion of design

considerations and cryptanalysis and introduces the Feistel cipher, which is

the basic structure of most modern symmetric encryption schemes.

**Chapter 4: Finite Fields**

Finite fields have become increasingly important in cryptography. A number of

cryptographic algorithms rely heavily on properties of finite fields, notably the

Advanced Encryption Standard (AES) and elliptic curve cryptography. This

chapter is positioned here so that concepts relevant to AES can be introduced

prior to the discussion of AES. Chapter 4 provides the necessary background

to the understanding of arithmetic over finite fields of the form GF(2*n*).

**Chapter 5: Advanced Encryption Standard**

The most important development in cryptography in recent years is the

adoption of a new symmetric cipher standard, AES. Chapter 5 provides a

thorough discussion of this cipher.

**Chapter 6: More on Symmetric Ciphers**

Chapter 6 explores additional topics related to symmetric ciphers. The chapter

begins by examining multiple encryption and, in particular, triple DES. Next,

we look at the concept of block cipher modes of operation, which deal with

ways of handling plaintext longer than a single block. Finally, the chapter

discusses stream ciphers and describes RC4.

**Chapter 7: Confidentiality Using Symmetric Encryption**

Beyond questions dealing with the actual construction of a symmetric

encryption algorithm, a number of design issues relate to the use of

symmetric encryption to provide confidentiality. Chapter 7 surveys the most

important of these issues. The chapter includes a discussion of end-to-end

versus link encryption, techniques for achieving traffic confidentiality, and key

distribution techniques. An important related topic, random number

generation, is also addressed.







[Page 28]

**Chapter 2. Classical Encryption Techniques**

**2.1 Symmetric Cipher Model**

Cryptography

Cryptanalysis

**2.2 Substitution Techniques**

Caesar Cipher

Monoalphabetic Ciphers

Playfair Cipher

Hill Cipher

Polyalphabetic Ciphers

One-Time Pad

**2.3 Transposition Techniques**

**2.4 Rotor Machines**

**2.5 Steganography**

**2.6 Recommended Reading and Web Sites**

**2.7 Key Terms, Review Questions, and Problems**

Key Terms

Review Questions

Problems

[Page 29]

*Many savages at the present day regard their names as vital parts of themselves, and therefore take great pains to conceal their real names, lest these should give to evil disposed persons a handle by which to injure their owners*.

*The Golden Bough*, Sir James George Frazer

| **Key Points**  ● Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption. ● Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the ciphertext.  ● The two types of attack on an encryption algorithm are cryptanalysis, based on properties of the encryption algorithm, and brute-force, which involves trying all possible keys.  ● Traditional (precomputer) symmetric ciphers use substitution and/or transposition techniques. Substitution techniques map plaintext elements (characters, bits) into ciphertext elements. Transposition techniques systematically transpose the positions of plaintext elements.  ● Rotor machines are sophisticated precomputer hardware devices that use substitution techniques.  ● Steganography is a technique for hiding a secret message within a larger one in such a way that others cannot discern the presence or contents of the hidden message. |
| --- |

Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the development of public-key encryption in the 1970s. It remains by far the most widely used of the two types of encryption. Part One examines a number of symmetric ciphers. In this chapter, we begin with a look at a general model for the symmetric encryption process; this will enable us to understand the context within which the algorithms are used. Next, we examine a variety of algorithms in use before the computer era. Finally, we look briefly at a different approach known as steganography. Chapter 3 examines the most widely used symmetric cipher: DES.

Before beginning, we define some terms. An original message is known as the **plaintext**, while the coded message is called the **ciphertext**. The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**; restoring the plaintext from the ciphertext is **deciphering** or **decryption**. The many schemes used for encryption constitute the area of study known as **cryptography**. Such a scheme is known as a **cryptographic system** or a **cipher**. Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**. Cryptanalysis is what the layperson calls "breaking the code." The areas of cryptography and cryptanalysis together are called **cryptology**.

[Page 30]





[Page 30 (continued)]

**2.1. Symmetric Cipher Model**

A symmetric encryption scheme has five ingredients (Figure 2.1):

● **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input. ● **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.

● **Secret key:** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

● **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.

● **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

**Figure 2.1. Simplified Model of Conventional Encryption**

[View full size image]



There are two requirements for secure use of conventional encryption:

**1.**

We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.

[Page 31]

Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

We assume that it is impractical to decrypt a message on the basis of the ciphertext *plus* knowledge of the encryption/decryption algorithm. In other words, we do not need to keep the algorithm secret; we need to keep only the key secret. This feature of symmetric encryption is what makes it feasible for widespread use. The fact that the algorithm need not be kept secret means that manufacturers can and have developed low-cost chip implementations of data encryption algorithms. These chips are widely available and incorporated into a number of products. With the use of symmetric encryption, the principal security problem is maintaining the secrecy of the key.

Let us take a closer look at the essential elements of a symmetric encryption scheme, using Figure 2.2. A source produces a message in plaintext, *X* = [*X*1, *X*2, ..., *XM*]. The *M* elements of *X* are letters in some finite alphabet. Traditionally, the alphabet usually consisted of the 26 capital letters. Nowadays, the binary alphabet {0, 1} is typically used. For encryption, a key of the form *K* = [*K*1, *K*2, ..., *KJ*] is generated. If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel. Alternatively, a third party could generate the key and securely deliver it to both source and destination.

**Figure 2.2. Model of Conventional Cryptosystem**

[View full size image]



With the message *X* and the encryption key *K* as input, the encryption algorithm forms the ciphertext *Y* = [*Y*1, *Y*2, ..., *YN*]. We can write this as

*Y* = E(*K*, *X*)

[Page 32]

This notation indicates that *Y* is produced by using encryption algorithm E as a function of the plaintext *X*, with the specific function determined by the value of the key *K*.

The intended receiver, in possession of the key, is able to invert the transformation: *X* = D(*K*, *Y*)

An opponent, observing *Y* but not having access to *K* or *X*, may attempt to recover *X* or *K* or both *X* and *K*. It is assumed that the opponent knows the encryption (E) and decryption (D) algorithms. If the opponent is interested in only this particular message, then the focus of the effort is to recover *X* by generating a plaintext estimate . Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover *K* by generating an estimate .

**Cryptography**

Cryptographic systems are characterized along three independent dimensions:

**1.**

**The type of operations used for transforming plaintext to ciphertext.** All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (that is, that all operations are reversible). Most systems, referred to as *product systems*, involve multiple stages of substitutions and transpositions.

**2.**

**The number of keys used.** If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.

**3.**

**The way in which the plaintext is processed.** A *block cipher* processes the input one block of elements at a time, producing an output block for each input block. A *stream cipher* processes the input elements continuously, producing output one element at a time, as it goes along.

**Cryptanalysis**

Typically, the objective of attacking an encryption system is to recover the key in use rather then simply to recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:

● **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to

deduce a specific plaintext or to deduce the key being used.

[Page 33]

● **Brute-force attack:** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

We first consider cryptanalysis and then discuss brute-force attacks.

Table 2.1 summarizes the various types of **cryptanalytic attacks**, based on the amount of information known to the cryptanalyst. The most difficult problem is presented when all that is available is the *ciphertext only*. In some cases, not even the encryption algorithm is known, but in general we can assume that the opponent does know the algorithm used for encryption. One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plaintext that is concealed, such as English or French text, an EXE file, a Java source listing, an accounting file, and so on.

**Table 2.1. Types of Attacks on Encrypted Messages**

| **Type of Attack** | **Known to Cryptanalyst** |
| --- | --- |
| Ciphertext only | ● Encryption algorithm  ● Ciphertext |
| Known plaintext | ● Encryption algorithm  ● Ciphertext  ● One or more plaintext-ciphertext pairs formed with the secret key |
| Chosen plaintext | ● Encryption algorithm  ● Ciphertext  ● Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key |
| Chosen ciphertext | ● Encryption algorithm  ● Ciphertext  ● Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |

| Chosen text | ● Encryption algorithm  ● Ciphertext  ● Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key  ● Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |
| --- | --- |

The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of information to work with. In many cases, however, the analyst has more information. The analyst may be able to capture one or more plaintext messages as well as their encryptions. Or the analyst may know that certain plaintext patterns will appear in a message. For example, a file that is encoded in the Postscript format always begins with the same pattern, or there may be a standardized header or banner to an electronic funds transfer message, and so on. All these are examples of *known plaintext*. With this knowledge, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed.

[Page 34]

Closely related to the known-plaintext attack is what might be referred to as a probable-word attack. If the opponent is working with the encryption of some general prose message, he or she may have little knowledge of what is in the message. However, if the opponent is after some very specific information,

then parts of the message may be known. For example, if an entire accounting file is being transmitted, the opponent may know the placement of certain key words in the header of the file. As another example, the source code for a program developed by Corporation X might include a copyright statement in some standardized position.

If the analyst is able somehow to get the source system to insert into the system a message chosen by the analyst, then a *chosen-plaintext* attack is possible. An example of this strategy is differential cryptanalysis, explored in Chapter 3. In general, if the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key.

Table 2.1 lists two other types of attack: chosen ciphertext and chosen text. These are less commonly employed as cryptanalytic techniques but are nevertheless possible avenues of attack.

Only relatively weak algorithms fail to withstand a ciphertext-only attack. Generally, an encryption algorithm is designed to withstand a known-plaintext attack.

Two more definitions are worthy of note. An encryption scheme is **unconditionally secure** if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. That is, no matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext, simply because the required information is not there. With the exception of a scheme known as the one-time pad (described later in this chapter), there is no encryption algorithm that is unconditionally secure. Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

● The cost of breaking the cipher exceeds the value of the encrypted information. ● The time required to break the cipher exceeds the useful lifetime of the information.

An encryption scheme is said to be **computationally secure** if either of the foregoing two criteria are

met. The rub is that it is very difficult to estimate the amount of effort required to cryptanalyze ciphertext successfully.

All forms of cryptanalysis for symmetric encryption schemes are designed to exploit the fact that traces of structure or pattern in the plaintext may survive encryption and be discernible in the ciphertext. This will become clear as we examine various symmetric encryption schemes in this chapter. We will see in Part Two that cryptanalysis for public-key schemes proceeds from a fundamentally different premise, namely, that the mathematical properties of the pair of keys may make it possible for one of the two keys to be deduced from the other.

[Page 35]

A **brute-force attack** involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. Table 2.2 shows how much time is involved for various key spaces. Results are shown for four binary key sizes. The 56-bit key size is used with the DES (Data Encryption Standard) algorithm, and the 168-bit key size is used for triple DES. The minimum key size specified for AES (Advanced Encryption Standard) is 128 bits. Results are also shown for what are called substitution codes that use a 26-character key (discussed later), in which all possible permutations of the 26 characters serve as keys. For each key size, the results are shown assuming that it takes 1 *m*s to perform a single decryption, which is a reasonable order of magnitude for today's machines. With the use of massively parallel organizations of microprocessors, it may be possible to achieve processing rates many orders of magnitude greater. The final column of Table 2.2 considers the results for a system that can process 1 million keys per microsecond. As you can see, at this performance level, DES can no longer be considered computationally secure.

**Table 2.2. Average Time Required for Exhaustive Key Search**

| **Key size (bits)** | **Number of**  **alternative keys** | | **Time required at 1**  **decryption/*m*s** | | **Time required at 106 decryption/*m*s** |
| --- | --- | --- | --- | --- | --- |
| 32 | 232 | = 4.3 x  109 | 231 *m*s | = 35.8 minutes | 2.15 milliseconds |
| 56 | 256 | = 7.2 x  1016 | 255 *m*s | = 1142 years | 10.01 hours |
| 128 | 2128 | = 3.4 x  1038 | 2127 *m*s | = 5.4 x 1024  years | 5.4 x 1018 years |
| 168 | 2168 | = 3.7 x  1050 | 2167 *m*s | = 5.9 x 1036  years | 5.9 x 1030 years |
| 26 characters  (permutation) | 26! | = 4 x 1026 | 2 x 1026  *m*s | = 6.4 x 1012  years | 6.4 x 106 years |







[Page 35 (continued)]

**2.2. Substitution Techniques**

In this section and the next, we examine a sampling of what might be called classical encryption techniques. A study of these techniques enables us to illustrate the basic approaches to symmetric encryption used today and the types of cryptanalytic attacks that must be anticipated.

The two basic building blocks of all encryption techniques are substitution and transposition. We examine these in the next two sections. Finally, we discuss a system that combines both substitution and transposition.

A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.[1]If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

[1] When letters are involved, the following conventions are used in this book. Plaintext is always in lowercase; ciphertext is in uppercase; key values are in italicized lowercase.

[Page 36]

**Caesar Cipher**

The earliest known use of a substitution cipher, and the simplest, was by Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet. For example,

plain: meet me after the toga party

cipher: PHHW PH DIWHU WKH WRJD SDUWB

Note that the alphabet is wrapped around, so that the letter following Z is A. We can define the transformation by listing all possibilities, as follows:

plain: a b c d e f g h i j k l m n o p q r s t u v w x y z

cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Let us assign a numerical equivalent to each letter:

| a | b | c | d | e | f | g | h | i | j | k | l | m |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| n | o | p | q | r | s | t | u | v | w | x | y | z |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Then the algorithm can be expressed as follows. For each plaintext letter *p*, substitute the ciphertext letter *C*:[2]

[2] We define *a* mod *n* to be the remainder when *a* is divided by *n*. For example, 11 mod 7 = 4. See Chapter 4 for a further discussion of modular arithmetic.

*C* = E(3, *p*) = (*p* + 3) mod 26

A shift may be of any amount, so that the general Caesar algorithm is

*C* = E(*k*, *p*) = (*p* + *k*) mod 26

where *k* takes on a value in the range 1 to 25. The decryption algorithm is simply *p* = D(*k*, *C*) = (*C k*) mod 26

If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed: Simply try all the 25 possible keys. Figure 2.3 shows the results of applying this strategy to the example ciphertext. In this case, the plaintext leaps out as occupying the third line.

**Figure 2.3. Brute-Force Cryptanalysis of Caesar Cipher**

**(This item is displayed on page 37 in the print version)**

****

Three important characteristics of this problem enabled us to use a brute-force cryptanalysis: **1.**

The encryption and decryption algorithms are known.

**2.**

There are only 25 keys to try.

**3.**

The language of the plaintext is known and easily recognizable.

[Page 37]

In most networking situations, we can assume that the algorithms are known. What generally makes brute-force cryptanalysis impractical is the use of an algorithm that employs a large number of keys. For example, the triple DES algorithm, examined in Chapter 6, makes use of a 168-bit key, giving a key space of 2168 or greater than 3.7 x 1050 possible keys.

The third characteristic is also significant. If the language of the plaintext is unknown, then plaintext output may not be recognizable. Furthermore, the input may be abbreviated or compressed in some fashion, again making recognition difficult. For example, Figure 2.4 shows a portion of a text file compressed using an algorithm called ZIP. If this file is then encrypted with a simple substitution cipher (expanded to include more than just 26 alphabetic characters), then the plaintext may not be recognized when it is uncovered in the brute-force cryptanalysis.

[Page 38]

**Figure 2.4. Sample of Compressed Text**

****

**Monoalphabetic Ciphers**

With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. Recall the assignment for the Caesar cipher:

plain: a b c d e f g h i j k l m n o p q r s t u v w x y z

cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

If, instead, the "cipher" line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4 x 1026 possible keys. This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Such an approach is referred to as a **monoalphabetic substitution cipher**, because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.

There is, however, another line of attack. If the cryptanalyst knows the nature of the plaintext (e.g., noncompressed English text), then the analyst can exploit the regularities of the language. To see how such a cryptanalysis might proceed, we give a partial example here that is adapted from one in [SINK66]. The ciphertext to be solved is

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ

VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX

EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English, such as is shown in Figure 2.5 (based on [LEWA00]). If the message were long enough, this technique alone might be sufficient, but because this is a relatively short message, we cannot expect an exact match. In any case, the relative frequencies of the letters in the ciphertext (in percentages) are as follows:

| P 13.33 | H 5.83 | F 3.33 | B 1.67 | C 0.00 |
| --- | --- | --- | --- | --- |
| Z 11.67 | D 5.00 | W 3.33 | G 1.67 | K 0.00 |
| S 8.33 | E 5.00 | Q 2.50 | Y 1.67 | L 0.00 |
| U 8.33 | V 4.17 | T 2.50 | I 0.83 | N 0.00 |
| O 7.50 | X 4.17 | A 1.67 | J 0.83 | R 0.00 |
| M 6.67 |  |  |  |  |

[Page 39]

**Figure 2.5. Relative Frequency of Letters in English Text**

[View full size image]



Comparing this breakdown with Figure 2.5, it seems likely that cipher letters P and Z are the equivalents of plain letters e and t, but it is not certain which is which. The letters S, U, O, M, and H are all of relatively high frequency and probably correspond to plain letters from the set {a, h, i, n, o, r, s}.The letters with the lowest frequencies (namely, A, B, G, Y, I, J) are likely included in the set {b, j, k, q, v, x, z}.

There are a number of ways to proceed at this point. We could make some tentative assignments and start to fill in the plaintext to see if it looks like a reasonable "skeleton" of a message. A more systematic approach is to look for other regularities. For example, certain words may be known to be in the text. Or we could look for repeating sequences of cipher letters and try to deduce their plaintext equivalents.

A powerful tool is to look at the frequency of two-letter combinations, known as digrams. A table similar to Figure 2.5 could be drawn up showing the relative frequency of digrams. The most common such digram is th. In our ciphertext, the most common digram is ZW, which appears three times. So we make the correspondence of Z with t and W with h. Then, by our earlier hypothesis, we can equate P with e. Now notice that the sequence ZWP appears in the ciphertext, and we can translate that sequence as "the." This is the most frequent trigram (three-letter combination) in English, which seems to indicate that we are on the right track.

Next, notice the sequence ZWSZ in the first line. We do not know that these four letters form a complete word, but if they do, it is of the form th\_t. If so, S equates with a.

[Page 40]

So far, then, we have

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ

t a e e te a that e e a a

VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX

e t ta t ha e ee a e th t a

EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

e e e tat e the t

Only four letters have been identified, but already we have quite a bit of the message. Continued analysis of frequencies plus trial and error should easily yield a solution from this point. The complete plaintext, with spaces added between words, follows:

it was disclosed yesterday that several informal but

direct contacts have been made with political

representatives of the viet cong in moscow

Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet. A countermeasure is to provide multiple substitutes, known as homophones, for a single letter. For example, the letter e could be assigned a number of different cipher symbols, such as 16, 74, 35, and 21, with each homophone used in rotation, or randomly. If the number of symbols assigned to each letter is proportional to the relative frequency of that letter, then single-letter frequency information is completely obliterated. The great mathematician Carl Friedrich Gauss believed that he had devised an unbreakable cipher using homophones. However, even with homophones, each element of plaintext affects only one element of ciphertext, and multiple-letter patterns (e.g., digram frequencies) still survive in the ciphertext, making cryptanalysis relatively straightforward.

Two principal methods are used in substitution ciphers to lessen the extent to which the structure of the plaintext survives in the ciphertext: One approach is to encrypt multiple letters of plaintext, and the other is to use multiple cipher alphabets. We briefly examine each.

**Playfair Cipher**

The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams.[3]

[3] This cipher was actually invented by British scientist Sir Charles Wheatstone in 1854, but it bears the name of his friend Baron Playfair of St. Andrews, who championed the cipher at the British foreign office.

[Page 41]

The Playfair algorithm is based on the use of a 5 x 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers's *Have His Carcase*:[4]

[4] The book provides an absorbing account of a probable-word attack.

| M | O | N | A | R |
| --- | --- | --- | --- | --- |
| C | H | Y | B | D |

| E | F | G | I/J | K |
| --- | --- | --- | --- | --- |
| L | P | Q | S | T |
| U | V | W | X | Z |

In this case, the keyword is *monarchy*. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter. Plaintext is encrypted two letters at a time, according to the following rules:

**1.** Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.

**2.** Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.

**3.** Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.

**4.** Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 x 26 = 676 digrams, so that identification of individual digrams is more difficult. Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult. For these reasons, the Playfair cipher was for a long time considered unbreakable. It was used as the standard field system by the British Army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

Despite this level of confidence in its security, the Playfair cipher is relatively easy to break because it still leaves much of the structure of the plaintext language intact. A few hundred letters of ciphertext are generally sufficient.

One way of revealing the effectiveness of the Playfair and other ciphers is shown in Figure 2.6, based on [SIMM93]. The line labeled *plaintext* plots the frequency distribution of the more than 70,000 alphabetic characters in the *Encyclopaedia Brittanica* article on cryptology.[5]This is also the frequency distribution of any monoalphabetic substitution cipher. The plot was developed in the following way: The number of

occurrences of each letter in the text was counted and divided by the number of occurrences of the letter e (the most frequently used letter). As a result, e has a relative frequency of 1, t of about 0.76, and so on. The points on the horizontal axis correspond to the letters in order of decreasing frequency.

[5] I am indebted to Gustavus Simmons for providing the plots and explaining their method of construction.

[Page 42]

**Figure 2.6. Relative Frequency of Occurrence of Letters**

[View full size image]



Figure 2.6 also shows the frequency distribution that results when the text is encrypted using the Playfair cipher. To normalize the plot, the number of occurrences of each letter in the ciphertext was again divided by the number of occurrences of e in the plaintext. The resulting plot therefore shows the extent to which the frequency distribution of letters, which makes it trivial to solve substitution ciphers, is masked by encryption. If the frequency distribution information were totally concealed in the encryption process, the ciphertext plot of frequencies would be flat, and cryptanalysis using ciphertext only would be effectively impossible. As the figure shows, the Playfair cipher has a flatter distribution than does plaintext, but nevertheless it reveals plenty of structure for a cryptanalyst to work with.

**Hill Cipher[6]**

[6] This cipher is somewhat more difficult to understand than the others in this chapter, but it illustrates an important point about cryptanalysis that will be useful later on. This subsection can be skipped on a first reading.

Another interesting multiletter cipher is the Hill cipher, developed by the mathematician Lester Hill in 1929. The encryption algorithm takes *m* successive plaintext letters and substitutes for them *m* ciphertext letters. The substitution is determined by *m* linear equations in which each character is assigned a numerical value (a = 0, b = 1 ... z = 25). For *m* = 3, the system can be described as follows:

[Page 43]

c1 = (*k*11*P*1 + *k*12*P*2 + *k*13*P*3) mod 26

c2 = (*k*21*P*1 + *k*22*P*2 + *k*23*P*3) mod 26

c3 = (*k*31*P*1 + *k*32*P*2 + *k*33*P*3) mod 26

This can be expressed in term of column vectors and matrices:



or

**C** = **KP** mod 26

where **C** and **P** are column vectors of length 3, representing the plaintext and ciphertext, and **K** is a 3 x 3 matrix, representing the encryption key. Operations are performed mod 26.

For example, consider the plaintext "paymoremoney" and use the encryption key 

The first three letters of the plaintext are represented by the vector



the ciphertext for the entire plaintext is LNSHDLEWMTRW.

Decryption requires using the inverse of the matrix **K**. The inverse **K**1 of a matrix **K** is defined by the equation **KK**1 = **K**1**K** = **I**, where **I** is the matrix that is all zeros except for ones along the main diagonal from upper left to lower right. The inverse of a matrix does not always exist, but when it does, it satisfies the preceding equation. In this case, the inverse is:



This is demonstrated as follows:



It is easily seen that if the matrix **K**1 is applied to the ciphertext, then the plaintext is recovered. To explain how the inverse of a matrix is determined, we make an exceedingly brief excursion into linear algebra.[7]For any square matrix (*m* x *m*) the **determinant** equals the sum of all the products that can be formed by taking exactly one element from each row and exactly one element from each column, with certain of the product terms preceded by a minus sign. For a 2 x 2 matrix

[7] The basic concepts of linear algebra are summarized in the Math Refresher document at the Computer Science Student Resource site at WilliamStallings.com/StudentSupport.html. The interested reader may consult any text on linear algebra for greater detail.

[Page 44]



the determinant is *k*11*k*22 *k*12*k*21. For a 3 x 3 matrix, the value of the determinant is *k*11*k*22*k*33 + *k*21*k*32*k*13 + *k*31*k*12*k*23 *k*31*k*22*k*13 *k*21*k*12*k*33 *k*11*k*32*k*23. If a square matrix **A** has a nonzero determinant, then the inverse of the matrix is computed as [**A**1]*ij* = (1)*i*+*j*(D*ij*)/ded(**A**), where (D*ij*) is the

subdeterminant formed by deleting the *i*th row and the *j*th column of **A** and det(**A**) is the determinant of **A**. For our purposes, all arithmetic is done mod 26.

In general terms, the Hill system can be expressed as follows:

**C** = E(**K, P**) = **KP** mod 26

**P** = D(**K, P**) = **K**1**C** mod 26 = **K**1**KP** = **P**

As with Playfair, the strength of the Hill cipher is that it completely hides single-letter frequencies. Indeed, with Hill, the use of a larger matrix hides more frequency information. Thus a 3 x 3 Hill cipher hides not only single-letter but also two-letter frequency information.

Although the Hill cipher is strong against a ciphertext-only attack, it is easily broken with a known plaintext attack. For an *m* x *m* Hill cipher, suppose we have *m* plaintext-ciphertext pairs, each of length *m*. We label the pairs



unknown key matrix **K**. Now define two *m* x *m* matrices **X** = (*Pij*) and **Y** = (*Cij*). Then we can form the matrix equation **Y** = **KX**. If **X** has an inverse, then we can determine **K** = **YX**1. If **X** is not invertible, then a new version of **X** can be formed with additional plaintext-ciphertext pairs until an invertible **X** is obtained.

We use an example based on one in [STIN02]. Suppose that the plaintext "friday" is encrypted using a 2 x 2 Hill cipher to yield the ciphertext PQCFKU. Thus, we know that



Using the first two plaintext-ciphertext pairs, we have



[Page 45]

The inverse of **X** can be computed:



so



This result is verified by testing the remaining plaintext-ciphertext pair.

**Polyalphabetic Ciphers**

Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic

substitutions as one proceeds through the plaintext message. The general name for this approach is **polyalphabetic substitution cipher**. All these techniques have the following features in common:

**1.**

A set of related monoalphabetic substitution rules is used.

**2.**

A key determines which particular rule is chosen for a given transformation.

The best known, and one of the simplest, such algorithm is referred to as the Vigenère cipher. In this scheme, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers, with shifts of 0 through 25. Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter a. Thus, a Caesar cipher with a shift of 3 is denoted by the key value *d*.

To aid in understanding the scheme and to aid in its use, a matrix known as the Vigenère tableau is constructed (Table 2.3). Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. A normal alphabet for the plaintext runs across the top. The process of encryption is simple: Given a key letter *x* and a plaintext letter y, the ciphertext letter is at the intersection of the row labeled *x* and the column labeled y; in this case the ciphertext is V.

**Table 2.3. The Modern Vigenère Tableau**

**(This item is displayed on page 46 in the print version)**

[View full size image]



To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword. For example, if the keyword is *deceptive*, the message "we are discovered save yourself" is encrypted as follows:

key: *deceptivedeceptivedeceptive*

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Decryption is equally simple. The key letter again identifies the row. The position of the ciphertext letter in that row determines the column, and the plaintext letter is at the top of that column.

The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. Thus, the letter frequency information is obscured. However, not all knowledge of the plaintext structure is lost. For example, Figure 2.6 shows the frequency distribution for a Vigenère cipher with a keyword of length 9. An improvement is achieved over the Playfair cipher, but considerable frequency information remains.

[Page 47]

It is instructive to sketch a method of breaking this cipher, because the method reveals some of the mathematical principles that apply in cryptanalysis.

First, suppose that the opponent believes that the ciphertext was encrypted using either monoalphabetic substitution or a Vigenère cipher. A simple test can be made to make a determination. If a monoalphabetic substitution is used, then the statistical properties of the ciphertext should be the same as that of the language of the plaintext. Thus, referring to Figure 2.5, there should be one cipher letter with a relative frequency of occurrence of about 12.7%, one with about 9.06%, and so on. If only a single message is available for analysis, we would not expect an exact match of this small sample with the statistical profile of the plaintext language. Nevertheless, if the correspondence is close, we can assume a monoalphabetic substitution.

If, on the other hand, a Vigenère cipher is suspected, then progress depends on determining the length of the keyword, as will be seen in a moment. For now, let us concentrate on how the keyword length can be determined. The important insight that leads to a solution is the following: If two identical sequences of plaintext letters occur at a distance that is an integer multiple of the keyword length, they will generate identical ciphertext sequences. In the foregoing example, two instances of the sequence "red" are separated by nine character positions. Consequently, in both cases, r is encrypted using key letter *e*, e is encrypted using key letter *p*, and d is encrypted using key letter *t*. Thus, in both cases the ciphertext sequence is VTW.

An analyst looking at only the ciphertext would detect the repeated sequences VTW at a displacement of 9 and make the assumption that the keyword is either three or nine letters in length. The appearance of VTW twice could be by chance and not reflect identical plaintext letters encrypted with identical key letters. However, if the message is long enough, there will be a number of such repeated ciphertext sequences. By looking for common factors in the displacements of the various sequences, the analyst should be able to make a good guess of the keyword length.

Solution of the cipher now depends on an important insight. If the keyword length is *N*, then the cipher, in effect, consists of *N* monoalphabetic substitution ciphers. For example, with the keyword DECEPTIVE, the letters in positions 1, 10, 19, and so on are all encrypted with the same monoalphabetic cipher. Thus, we can use the known frequency characteristics of the plaintext language to attack each of the monoalphabetic ciphers separately.

The periodic nature of the keyword can be eliminated by using a nonrepeating keyword that is as long as the message itself. Vigenère proposed what is referred to as an **autokey system**, in which a keyword is concatenated with the plaintext itself to provide a running key. For our example,

key: *deceptivewearediscoveredsav*

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA

Even this scheme is vulnerable to cryptanalysis. Because the key and the plaintext share the same frequency distribution of letters, a statistical technique can be applied. For example, e enciphered by *e*, by Figure 2.5, can be expected to occur with a frequency of (0.127)2 0.016, whereas t enciphered by *t* would occur only about half as often. These regularities can be exploited to achieve successful cryptanalysis.[8]

[8] Although the techniques for breaking a Vigenère cipher are by no means complex, a 1917 issue of *Scientific American* characterized this system as "impossible of translation." This is a point worth remembering when similar claims are made for modern algorithms.

[Page 48]

The ultimate defense against such a cryptanalysis is to choose a keyword that is as long as the plaintext and has no statistical relationship to it. Such a system was introduced by an AT&T engineer named Gilbert Vernam in 1918. His system works on binary data rather than letters. The system can be expressed succinctly as follows:

*ci* = *pi ki*

where

*pi* = *i*th binary digit of plaintext

*ki* = *i*th binary digit of key

*ci* = *i*th binary digit of ciphertext

= exclusive-or (XOR) operation

Thus, the ciphertext is generated by performing the bitwise XOR of the plaintext and the key. Because of the properties of the XOR, decryption simply involves the same bitwise operation:

*pi* = *ci ki*

The essence of this technique is the means of construction of the key. Vernam proposed the use of a running loop of tape that eventually repeated the key, so that in fact the system worked with a very long but repeating keyword. Although such a scheme, with a long key, presents formidable cryptanalytic

difficulties, it can be broken with sufficient ciphertext, the use of known or probable plaintext sequences, or both.

**One-Time Pad**

An Army Signal Corp officer, Joseph Mauborgne, proposed an improvement to the Vernam cipher that yields the ultimate in security. Mauborgne suggested using a random key that is as long as the message, so that the key need not be repeated. In addition, the key is to be used to encrypt and decrypt a single message, and then is discarded. Each new message requires a new key of the same length as the new message. Such a scheme, known as a **one-time pad**, is unbreakable. It produces random output that bears no statistical relationship to the plaintext. Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code.

An example should illustrate our point. Suppose that we are using a Vigenère scheme with 27 characters in which the twenty-seventh character is the space character, but with a one-time key that is as long as the message. Thus, the tableau of Table 2.3 must be expanded to 27 x 27. Consider the ciphertext

ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

[Page 49]

We now show two different decryptions using two different keys:

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key: *pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih*

plaintext: mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key: *mfugpmiydgaxgoufhklllmhsqdqogtewbqfgyovuhwt*

plaintext: miss scarlet with the knife in the library

Suppose that a cryptanalyst had managed to find these two keys. Two plausible plaintexts are produced. How is the cryptanalyst to decide which is the correct decryption (i.e., which is the correct key)? If the actual key were produced in a truly random fashion, then the cryptanalyst cannot say that one of these two keys is more likely than the other. Thus, there is no way to decide which key is correct and therefore which plaintext is correct.

In fact, given any plaintext of equal length to the ciphertext, there is a key that produces that plaintext. Therefore, if you did an exhaustive search of all possible keys, you would end up with many legible plaintexts, with no way of knowing which was the intended plaintext. Therefore, the code is unbreakable.

The security of the one-time pad is entirely due to the randomness of the key. If the stream of characters that constitute the key is truly random, then the stream of characters that constitute the ciphertext will be truly random. Thus, there are no patterns or regularities that a cryptanalyst can use to attack the ciphertext.

In theory, we need look no further for a cipher. The one-time pad offers complete security but, in practice, has two fundamental difficulties:

**1.**

There is the practical problem of making large quantities of random keys. Any heavily used system might require millions of random characters on a regular basis. Supplying truly random characters in this volume is a significant task.

**2.**

Even more daunting is the problem of key distribution and protection. For every message to be sent, a key of equal length is needed by both sender and receiver. Thus, a mammoth key distribution problem exists.

Because of these difficulties, the one-time pad is of limited utility, and is useful primarily for low bandwidth channels requiring very high security.





[Page 49 (continued)]

**2.3. Transposition Techniques**

All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.

The simplest such cipher is the rail fence technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows. For example, to encipher the message "meet me after the toga party" with a rail fence of depth 2, we write the following:

[Page 50]

m e m a t r h t g p r y

e t e f e t e o a a t

The encrypted message is

MEMATRHTGPRYETEFETEOAAT

This sort of thing would be trivial to cryptanalyze. A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. For example,

Key: 4 3 1 2 5 6 7

Plaintext: a t t a c k p

o s t p o n e

d u n t i l t

w o a m x y z

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. For the type of columnar transposition just shown, cryptanalysis is fairly straightforward and involves laying out the ciphertext in a matrix and playing around with column positions. Digram and trigram frequency tables can be useful.

The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easily reconstructed. Thus, if the foregoing message is reencrypted using the same algorithm,

Key: 4 3 1 2 5 6 7

Input: t t n a a p t

m t s u o a o

d w c o i x k

n l y p e t z

Output: NSCYAUOPTTWLTMDNAOIEPAXTTOKZ